Appendix H – Regular Expressions in (L)INSTR

The alternate forms of the INSTR() and LINSTR() functions can take a regular expression as the search pattern.

The alternate form of the functions are:

INSTR([start],text$, search$ [,size])

LINSTR(text%(),search$ [,start] [,size])

In both cases specifying the size parameter causes the firmware to interpret the search string as a regular expression. The size parameter is a floatingpoint variable that is used by the firmware to return the size of a matching string. If the variable doesn't exist it is created. As implemented in MMBasic you need to apply the returned ***start*** and ***size*** values to the **MID$** function to extract the matched string. e.g.

IF start THEN match$=MID$(text$,start,size) ELSE match$="" ENDIF

The syntax of regular expressions can vary slightly with the various implementations. This appendix is a summary of the syntax and supported operations used in the MMBasic implementation on the ArmmiteH7

Anchors		**Quantifiers**		**Groups and Ranges**
^ Start of string		* 0 or more		(a|b|c) a or b or c
$ End of string		+ 1 or more		(…) group
\b Word Boundary (space,or hyphen)		? 0 or 1		[abc] Range (a or b or c)
\B Not a word boundary		{3} Exactly 3		[^abc] Not (a or b or c]
~~\< Start of word~~		{3,} 3 or more		[a-q] lower case letters a to q
~~\> End of word~~		{,5} 5 or less		[A-Q] upper case letters A to Q
		{3,5} 3,4 or 5		[0-7] Digits from 0 to 7
Escapes required to match Special Characters		Escapes with special functions		Character Classes
\^ to match ^ (caret)				\w digits and letters plus _
\. to match . (dot)				\W non -alpha
* to match * (asterix)				\s white space \t \f \r \n \v spaces
\$ to match $ (dollar)				\S non white space
\[to match [(left bracket)				\d digits
\\ to match \ (backslash)				\D non digits
\? to match ? (questmark)				\xXX hex encoded byte
\(to match (				
\) to match)				Matching rules
\+ to match + (plus)				non special chars match themselves
| to match pipe				. (dot) matches any character

Limitations.

Anchors within a group are not supported.

e.g. (^hello) or (hello$) will not match "hello" at beginning or end of the line as expected.

However, anchors outside the group are ok. e.g. ^(hello) or (hello)$ will match.

Example expression to match an IP Address.

"[\d]+\.[\d]+\.[\d]+\.[\d]+"