**Available assembler directives.**

| | |
|---|---|
| ORG | **Set location counter to ORiGin.**<br>`START  ORG  $E000`<br>This allows the user to set the location counter to a specified value. This is commonly used to ensure that code is assembled for the correct addresses which correspond to your memory map. The above example sets the location counter to the address $E000 which is the beginning of ROM in some MC68HC11 microcontrollers. |
| FCC | **Form Constant Character string.**<br>`STRING FCC  'A string of ASCII'`<br>This causes the label to be assigned the address of the first letter in the string. The string is stored in successive bytes with its ASCII value. Instead of double quotes single quotes can be used. |
| FCB | **Form Constant Byte.**<br>`TABLE  FCB  0,1,$02,'A'`<br>This causes the assembler to store the operands in successive 8-bit bytes. The operands must be 8-bit values or single character constants. Each value is separated by a comma. |
| FDB | **Form Double Byte.**<br>`CONSTS FDB  0000,$1234,"!!`<br>Function is similar to that of FCB except 16-bit values are stored sequentially in memory. These values may be numbers or double ASCII. |
| EQU | **EQUate symbol to a value.**<br>`LABEL  EQU  $1017`<br>This causes the assembler to add the label to the symbol table and equates it to the given value. In this example, LABEL equates to $1000. The label cannot be redefined elsewhere. The value must not be forward-referenced or undefined. |
| #INCLUDE | **Include an other source file.**<br>`#INCLUDE "file.inc"`<br>This causes the assembler to read in the specified file. This is often used to read in a lot of "standard" EQU's. |
| RMB | **Reserve Memory Bytes.**<br>`VARS   RMB  3`<br>This directive allows the user to reserve addresses and associate a label to that address for variables, tables, etc. This example reserves the next three sequential bytes for variables and associates the first address with the symbol VARS. |
| END | **End of assembler input.**<br>`END`<br>This causes the assembler to stop reading the file. All lines following this line are comments. If you do not use an END directive the assembler reads until the end of file. |