IDE11 V2.30 - Integrated Development Environment for 68HC11

This file contains a short overview and description of the most important IDE11 features. Information on software installation and connection of target hardware can be found in the README.DOC.

Assembler Directives:

Directives are operations that do not generate instructions for the CPU but affect the assembler in certain ways. The following list contains the most often used assembler directives for IDE11 (always shown as an example):

```
        ORG $B600
```

Sets the program counter to the specified address. Default address is $0000. Multiple ORGs within one program must appear in ascending order.

```
name1   EQU $80
name2   EQU name1 + 1
```

Assigns a value to the label "name1". Note: Labels are case sensitive by default, so "name1" and "NAME1" are different labels!

```
label1  DB  $A0
        DB  1,2,3,4,5,6
label2  DB  "Hello message"
label3  DB  "Hello", $20, "message", 0
```

Allocates bytes and initializes them with the given values. Strings are allowed, even mixing strings and numeric values. Labels are optional. You may check the generated code by taking a look on the listing file.

```
label   DW  $E000
        DW  main
```

Same as DB but uses word values. Good for specifying interrupt vectors etc. The label is optional.

```
label   DL  $12345678
```

Same as DB but uses long values (4 byte, most significant byte first).

```
label   RMB 20
```

Reserves the specified number of bytes without assigning a value.
The label is optional.

```
label   RMW 20
```

Same as RMB but uses word values. The example reserves 20*2=40 bytes.

```
label   RML 20
```

Same as RMB but uses long values. The example reserves 20*4=80 bytes.

```
        INCLUDE "MYSTUFF.A"
        INCLUDE "C:\XYZ\ANOTHER.A"
        INCLUDE <HC11AE.H>
```

Includes the given file into the current source file. The line of
the current file will be replaced by the lines of the include file.
The assembler searches the file MYSTUFF.A in the current directory.
The assembler searches the file ANOTHER.A in the specified path.
The third version (use of <...>) tells the assembler to search in
the default include directory (as specified in the Options/Assembler
dialog box). You should set the standard include path option to the
directory which contains your include files (e.g. C:\IDE11\LIB).

```
        NOLIST
        LIST
```

Disables / enables the generation of the assembly listing for the
following source lines. Default: enabled.
Note: You must allow the creation of an assembly listing file by
checking the appropiate box in the Options/Assembler dialog.

Alternate Assembler Directives:

For compatibility reasons you may alternatively use the following

assembler directives:

Preferred	Alternatives
ORG	.ORG
EQU	.EQU
RMB	.DS DEFS DS DS.B
RMW	.DS.W
RML	.DS.L
DB	.DB DEFB DC.B BYTE .BYTE FCB STRING DCC FCC
DW	.DW DEFW DC.W WORD FDB
DL	DC.L
INCLUDE	.INCLUDE INCL LIB #INCLUDE (1)
LIST	.LIST
NOLIST	.NOLIST

Note (1): #INCLUDE must start in the first text column!

The following directives will be ignored (no error will be displayed):

BSS TEXT DATA END .END

Assembler directives are not case sensitive.

Comments:
■■■■■■■■■

Comments can start anywhere in the source line. They start with the ";" character or with a double slash "//".
Alternatively the "*" character is allowed provided the comment starts in the first (very left) text column. At all other positions "*" is considered to be a multiply operator.
Another way is to embrace a text passage in c-style comment, e.g.:

```
label  /* remark */  JSR  sub1
```

Labels:
■■■■■■■

Labels are case sensitive and have up to 31 significant characters. Allowed character: "a"-"z", "A"-"Z", "0"-"9", "@", "_" and ".". The first character must not be a number or ".". Labels can optionally end with a colon ":" (ignored by assembler).

Instructions:
■■■■■■■■■■■■■

Refer to the HC11 Reference Manual for details on instructions.
The INSTRUCT.DOC file contains a table with all HC11 instructions.

Addressing Modes:
■■■■■■■■■■■■■■■■

The notation of operands depending of the addressing mode is given

here by examples:

Addr. Mode	Examples
Immediate	#$20 #123 #'A' #isr_vec
Direct	<$0000 <VAR1
Extended	$A000 2400 main
Indexed	0,X 2,y
Relative	label1

Some confusion exists concerning the BSET/BCLR and BRSET/BRCLR
instructions. Here are some examples how to use them:

```
        BSET  0,x %00010000
        BCLR  0,y FLAG_BIT
        BSET  $2A $80

here    BRSET 0,x 1 here
wait    BRCLR 0,y $80 wait
        BRSET flag $FF main
```

This is identical to the description in the HC11 Reference Manual.

Constants:
==========

Numerical constants may be decimal (e.g. 1, 99, 800000), hexadecimal
(e.g. $2C, $8000, $9abcde) or binary (e.g. %00101100).
Character constants have the form 'A'. Example:

```
        LDAA #'z'
```

String constants look like this: "String constant". This can be used
in conjunction with a DB directive:

```
    DB  "Hello, world!"
```

Operators:
==========

Valid operators are: "+", "-", "*" and "/". Complex expressions may
be freely build using "(" and ")".

Information of Target Settings (Options/Target/Detail):
==========

--> MCU type

There are many different HC11 family members, it is necessary to tell
IDE11 which one is used. The only difference between 'A8, 'A1 and 'A0
is the presence of EEPROM and/or ROM: The 'A8 has EEPROM *and* ROM,
the 'A1 has EEPROM, but no ROM and the 'A0 has none of them. The same

scheme is used with 'E9, 'E1 and 'E0. The HC812E2 differs from 'A1 concerning size and location of EEPROM. It is the only member with 2KB on chip EEPROM. The F1 is an advanced family member with an unmultiplexed external address/data bus.

--> External memory

IDE11 can handle RAM and/or EEPROM as external memory. If your target operates in Expanded Multiplexed Mode (external memory connected) you must set the correct memory type, split into a lower and an upper memory area. Remark: You must ALWAYS set your target to Special Bootstrap Mode initially. If it is necessary to change to Expanded Mode, IDE11 will do it for you, depending on the setting:

--> Mode ctrl
    Default: None

Normally no (hardware) Mode Control is necessary, because IDE11 can use a software command to switch from Special Bootstrap Mode to Expanded Mode. But sometimes it is more realistic to switch the MODA+MODB pins. This can be done manually or using the output /RTS of the Host PC. Pay attention to the TTL/RS232 converter which is neccessary when using /RTS line.

--> Reset ctrl
    Default: /DTR=L

It is highly recommended to let IDE11 reset your target automatically when necessary. This (for example) happens when a download starts, when you access the CONFIG Register and at the moment a program is executed. You also can invert the reset line. If you choose the option /DTR=H the target will receive an active high reset pulse. See also comments on cabling in the README.DOC file.
If you choose the option Manual you will have to reset your target manually each time IDE11 asks you to do so.