
Freescall HC12 Assembler Directives

Section Definition

ORG Set program counter to the origin of the program in an absolute assembler mode.

SECTION Define a relocatable section.

OFFSET Define an offset section.

Constant Definition

EQU Equate symbol to an expression (cannot be redefined).

SET Assign a name to an expression (can be redefined).

Reserving or Allocating Memory Locations

DS Define Storage.

Defining Constants in Memory

DC.B Define byte constant.

DC.W Define word constant.

DCB Define a constant block.

RAD50 RAD50 encoded string constants.

Export or Import Global Symbols

ABSENTRY Specify the entry point in an absolute assembly file.

XDEF Make a symbol public (visible to some other file).

XREF Import reference to an external symbol.

XREFB Import reference to an external symbol located on the direct page.

Assembly Control

ALIGN Define alignment constraint.

BASE Specify default base for constants.

END End of assembly unit.

EVEN Define two byte alignment constraint.

FAIL Generate user defined error or warning messages.

INCLUDE Include text from another file.

LONGEVEN Define four byte alignment constraint

Repetitive Assembly Control

FOR Repeat assembly blocks.

ENDFOR End of FOR block.

Listing Control

CLIST Include conditional assembly block.

LIST Specify that all following assembly lines are in the list file.

LLEN Define line length.

MLIST Include macro expansions.

NOLIST Specify that all following assembly lines are not in the list file.

NOPAGE Disable pagination in the list file.

PAGE Insert page break.

PLEN Define page length.

SPC Insert empty or blank line.

TABS Define number of characters to insert for the <tab>.

TITLE User defined title.

Macro Definition

ENDM End of user defined macro.

MACRO Start of user defined macro.

MEXIT Exit from macro expansion.

Conditional Assembly

Freescale HC12 Assembler Directives

ELSE	Alternate block, code included if IF statement not true.
ENDIF	
IF	End of conditional block.
IFC	Start of conditional block.
IFDEF	Test if two string expressions are equal.
IFEQ	Test if a symbol is defined.
IFGE	Test if an expression is null.
IFGT	Test if an expression is greater than or equal to 0.
IFLE	Test if an expression is greater than 0.
IFLT	Test if an expression is less than or equal to 0.
IFNC	Test if an expression is less than 0.
IDNDEF	Test if two string expressions are different.
IFNE	Test if a symbol is undefined.
	Test if an expression is not null.

Base Designators for Constants

Base	Prefix
Binary (2)	%
Decimal (10)	none
Hexadecimal (16)	\$ (Default)
Octal	@

Assembler Expressions

+	Addition	-	Subtraction
*	Multiplication		
/	Division produces truncated result	%	Modulo division
>>	Shift right	<<	Shift left
&	Bitwise AND		Bitwise OR
^	Bitwise Exclusive OR (XOR)	~	1's complement
!= or <>	Logical NOT	= or ==	Equal
<=	Not equal	<	Less than
>=	Less than or equal	>	Greater than
HIGH	Greater than or equal	LOW	Low byte of an address
PAGE	High byte of an address		
	Page byte of an address		

Linker Parameter File for M68HC912B32 EVB

NAMES END

SECTIONS

```
uC_RAM = READ_WRITE      0x0800 TO 0x0AFF;
MyStack = READ_WRITE     0x0B00 TO 0x0BFF;
uC_FLASH = READ_ONLY     0x8000 TO 0xFFBF;
uC_EEPROM = READ_WRITE   0x0D00 TO 0x0FFF;
/* Define pseudo_ROM in the EVB RAM */
EVB_Pseudo_ROM = READ_ONLY 0x4000 TO 0x5FFF;
/* Define RAM in the EVB RAM */
EVB_RAM = READ_WRITE      0x6000 TO 0x7FFF;
END
```

PLACEMENT

```
/* Place the Code Segments */
.init, MainCode, SubCode INTO EVB_Pseudo_ROM;
/* Place the Constant Data sections */
MainConst, SubConst INTO EVB_Pseudo_ROM;
DEFAULT_ROM INTO EVB_Pseudo_ROM;
/* Place the Variable Data */
MyData, MySubData INTO EVB_RAM;
DEFAULT_RAM INTO EVB_RAM;
/* Place the Stack */
.stack INTO MyStack;
END
```

```
/* Specify the initial entry point */
INIT Entry
```