

Aufbau einer Programmzeile:

Allgemein:

Marke (Label):	Operationscode	Operanden	; Kommentar
----------------	----------------	-----------	-------------

Operationscode
Operationscode Operand
Operationscode Zieloperand, Quelloperand
Marke (Label):
; Kommentar

Assemblerdirektiven:

Bezeichnung	Wirkung	Beispiele
Einführen mnemonischer Bezeichnungen		
DEF	Benennung von Registern	.def time_count = r17
EQU	Benennung von Konstanten	.equ limit = 0x34
SET	Benennung von Konstanten	.set limit =
UNDEF	Benennung von Registern aufheben (zwecks erneuter Benennung)	.undef time_count
Segmentauswahl		
CSEG	die nachfolgenden Angaben betreffen das Programmsegment (Codesegment), also den Flash- ROM	.cseg
DSEG	die nachfolgenden Angaben betreffen das Datensegment, also den SRAM	.dseg
ESEG	die nachfolgenden Angaben betreffen den EEPROM	.eseg
Festlegen der Startadresse		
ORG	im betreffenden Segment wird die Startadresse zum Assemblieren der nachfolgenden Angaben eingestellt	.org 0

Bezeichnung	Wirkung	Beispiele
Platz für Variable reservieren (SRAM, EEPROM)		
BYTE	die angegebene Byteanzahl wird im betreffenden Segment (SRAM oder EEPROM) freigehalten. Der zugehörigen Marke wird die Anfangsadresse zugewiesen	string: <code>.byte 30</code> count: <code>.byte 2</code>
Konstanten definieren (Programmspeicher, EEPROM)		
DB, DW, DD, DQ	die angegebenen Konstanten werden auf das jeweilige Format gerundet und im betreffenden Segment (Programmspeicher oder EEPROM) abgelegt	<code>.db 0x23, 15,</code> <code>0b01101110,</code> <code>"Textbeispiel", 0</code> <code>.dw 0x23af</code> <code>.dd 0x1234567</code> <code>.dq 0x123456789abcdef</code>
Organisation und Programmentwicklung		
INCLUDE	Einbinden der angegebenen Quelldatei	<code>.include "m16def.inc"</code>
DEVICE	Zuweisung des Schaltkreistyps, für den der Code erzeugt werden soll	<code>.device atmega16</code>
MACRO	Beginn einer Makrodefinition	<code>.macro outi</code> <code>ldi r16, @1</code> <code>out @2, r16</code> <code>.endm</code>
ENDMACRO, ENDM	Ende einer Makrodefinition	<code>.endmacro</code>
LIST	die Übersetzung des nachfolgenden Quelltexts wird ins Listfile übernommen	<code>.list</code>
NOLIST	die Übersetzung des nachfolgenden Quelltexts wird nicht ins Listfile übernommen	<code>.nolist</code>
LISTMAC	die Übersetzung der Makros wird ins Listfile übernommen (ansonsten erscheint nur der Makroaufruf)	<code>.listmac</code>
EXIT	der nachfolgende Quelltext wird beim Assemblieren übergangen	<code>.exit</code>