AVR-8-bit-Mikrocontroller
Gruppe 200 - Einsetzen von AVR-Tools
Teil 207 - Editor - UltraEdit

Teil 201 - Experimentierboards

Teil 202 - ISP-Programmieradapter

Teil 203 - AVR-ALE-Testboard

Teil 204 - AVR Studio

Teil 205 - Assembler und AVR Studio

Teil 206 - C-Compiler und AVR Studio

Teil 207 - Editor - UltraEdit

Hinweis

Externe Anschaltungen und Hardware-Erweiterungen werden in der **Gruppe 400 - ASM-Projekte** und in der **Gruppe 600 - C-Projekte** detailliert beschrieben.

7 Editor - UltraEdit (Version 22.20.0.34)

Quell-Programme schreibt man mit einem Editor. Der braucht im Prinzip nicht mehr zu können, als ASCII-Zeichen zu schreiben und zu speichern. Im Prinzip täte es auch ein sehr einfaches Text-Programm. Da sowohl **AVR Studio** als auch der **CVAVR-Compiler** Editoren integriert haben, können diese auch hier eingesetzt werden. Wegen einer noch besseren Übersichtlichkeit und einer komfortablen Drucksteuerung wird jedoch der Allround-Editor **UltraEdit** verwendet. Eine Hilfe-Datei befindet sich hier. Dieser Editor bietet darüber hinaus die Möglichkeit, die verschiedensten Quellen mit einer erweiterten selbst definierbaren **Syntaxhervorhebung** zu editieren. Früher war das Coding für die Syntaxerweiterung in der Datei **WORDFILE.TXT** für alle verfügbaren Quellen enthalten. In der neuen Version von **UltraEdit** werden für jeden möglichen Quell-Code individuelle einzelne Dateien mit der Dateiendung ***.uew** in einem speziellen Ordner auf der Systemplatte eingesetzt. Dieser Ordner heißt **Wordfiles** und ist - sofern keine spezielle andere Einstellung vorgenommen wurde - in folgendem Systemordner angelegt:

C:\Benutzer*User-Name*\AppData\Roaming\IDMComp\UltraEdit\Wordfiles

Wer die "AppData" im Windows-Explorer nicht sieht, sollte ggf. die Einstellung im Explorer ändern. Das Häkchen bei folgender Einstellung ist aufzuheben:

Ansicht => Optionen => Ordneroptionen => Ansicht => Geschützte Systemdateien ausblenden (empfohlen)

Bereits bei der Installation von **UltraEdit** werden viele häufig gebrauchte "Syntaxhervorhebungs-Dateien" mit der Endung ***.uew** in diesem Ordner abgelegt. Doch wie es so ist, manchem fehlt immer noch "Dies oder Das", so dass **UltraEdit** eine Sammlung verschiedenster "Languages" zur Auswahl und zum Download anbietet:

https://www.ultraedit.com/downloads/extras/wordfiles.html

Wenn man daraus Dateien herunterlädt und sie in den o.a. Ordner platziert, werden sie beim nächsten Aufruf von **UltraEdit** automatisch zur Auswahl mit angezeigt. Wichtig ist es, dass die ***.uew-Dateien** wirklich in dem genannten Ordner und nicht in anderen Systemordnern wie z.B.:

C:\Program Files\IDM Computer Solutions\UltraEdit\wordfiles

oder

C:\Benutzer\info\AppData\Roaming\IDMComp\UltraEdit\wordfiles

kopiert werden. Warum hier ebenfalls scheinbar die gleichen *.uew-Dateien erscheinen, bleibt ein hier nicht lösbares Geheimnis des Software-Produzenten.

Wie noch weiter unten ausführlich beschrieben wird, werden für den Assembler **ASM** und den Compiler **CodeVisionAVR** (**CVAVR**) individuelle ***.uew-Dateien** konzipiert.

Dieser Editor hat insbesondere folgende Vorteile:

a) Er besitzt Zeilen- und Spalten-Nummerierung,
b) man kann beliebige Tabulator-Abstände setzen,
c) es kann die Schriftart und -größe frei gewählt werden (sinnvollerweise **Courier New; 10 pt**),
d) er bietet umfangreiche Such-, Ersetze- und Datei-Vergleichs-Funktionen,
e) er druckt verschiedene Assembler- oder CVAVR-Elemente in vordefinierten Farben aus,
b) er sieht eine sehr freie Gestaltung der Kopf- und Fußzeilen beim Ausdruck vor.

7.1 Kopf- und Fußzeile

Die Kopf- oder Fußzeile lässt sich mit dem entsprechenden Kontrollkästchen ein- bzw. ausschalten. Falls eine Kopf- und/oder Fußzeile aktiviert ist, kann in der Kopfzeile oberhalb bzw. in der Fußzeile unterhalb der Trennlinie optionaler Text beim Druck ausgegeben werden. Die Kopf- bzw. Fußzeile kann neben dem benutzerdefinierten Text zusätzlich Sonderzeichen enthalten, die das Drucken von Dateinamen, Seitenzahlen usw. ermöglichen.

Es gibt folgende Sonderzeichen um eine Druckseite einzurichten:

- **&f** Platzhalter in der Kopf-/Fußzeile für den vollen Dateinamen **mit** Pfad.
- **&n** Platzhalter in der Kopf-/Fußzeile für den Dateinamen **ohne** Pfad.
- **&p** Platzhalter in der Kopf-/Fußzeile für die aktuelle Seitenzahl.
- **&t** Platzhalter in der Kopf-/Fußzeile für die Gesamtanzahl der Seiten.
- **&l** Richtet den nachfolgenden Text **links** in der Kopf-/Fußzeile aus.
- **&c** Richtet den nachfolgenden Text **zentriert** in der Kopf-/Fußzeile aus.
- **&r** Richtet den nachfolgenden Text **rechts** in der Kopf-/Fußzeile aus.

Hinweis: Zeichen zum Ausrichten sind nicht erforderlich, können aber auch alle wahlweise in einer einzelnen Kopf-/Fußzeile auftreten. Wenn jedoch gleichzeitig mehrere Sonderzeichen verwendet werden sollen, muss die obige Reihenfolge eingehalten werden (z.B.: **&l** muss vor **&c** und **&c** muss vor **&r** stehen).

Zusätzlich können Kopf- oder Fußzeile das Datum und die Uhrzeit der Datei oder des Systems enthalten. Sonderzeichen legen fest, ob und welches Format gegebenenfalls verwendet werden soll. In der nachstehenden Tabelle stehen die Zeichen, mit denen die Formatierung erfolgt; das Fehlen von Formatierungszeichen bewirkt, dass Datum und Uhrzeit nicht gedruckt werden:

Format-Beschreibung

- **%a** Abgekürzter Wochentagsname
- **%A** Ausgeschriebener Wochentagsname
- **%b** Abgekürzter Monatsname
- **%B** Ausgeschriebener Monatsname
- **%c** Darstellung von Datum und Uhrzeit entsprechend der Ländereinstellung
- **%d** Tag des Monats als Dezimalzahl (01 bis 31)
- **%H** Stunden im **24-Stunden-Format** (00 bis 23)
- **%I** Stunden im **12-Stunden-Format** (01 bis 12)
- **%j** Tag des Jahres als Dezimalzahl (001 bis 366)
- **%m** Monat als Dezimalzahl (01 bis 12)
- **%M** Minuten als Dezimalzahlen (00 bis 59)
- **%p** Aktuelle Ländereinstellung für die **am/pm-Anzeige** bei 12-Stunden-Uhr
- **%S** Sekunden als Dezimalzahlen (00 bis 59)
- **%U** Woche des Jahres als Dezimalzahl mit Sonntag als erstem Tag der Woche (00 bis 51)
- **%w** Wochentag als Dezimalzahl (0 bis 6; Sonntag ist 0)
- **%W** Woche des Jahres als Dezimalzahl. Mit Montag als erstem Tag der Woche (00 bis 51)
- **%x** Datumsdarstellung für die aktuelle Ländereinstellung
- **%X** Zeitdarstellung für die aktuelle Ländereinstellung
- **%y** Jahr **ohne** Jahrhundert als Dezimalzahl (00 bis 99)
- **%Y** Jahr **mit** Jahrhundert als Dezimalzahl
- **%Z** Name der Zeitzone oder Abkürzung; keine Zeichen, falls die Zeitzone unbekannt ist
- **%%** Prozentzeichen

Beispiele (man beachte neben den Sonderzeichen die "normalen" Druckzeichen wie Komma, Leerzeichen und Uhr):

1. Systemzeit ist der 8. Juli 2009 um 9:00 Uhr abends (pm)

%X Uhr, %x	druckt	**21:00 Uhr, 08.07.09**
%c	druckt	**08.07.09 21:00**

2. Systemzeit ist der 10. April 2009 um 5:00 Uhr morgens (am)

%A, den %d. %B %Y, %I:%M%p	druckt	**Freitag, 10. April 2009, 05:00am**

7.1.1 Einstellungen für Assembler-Programme

Die bevorzugte Einstellungen für Assembler-Programme bezieht sich auf **DIN A4 Hochformat** und sollte wie folgt vorgenommen werden:

Datei => Druckeinstellungen => Seite einrichten...

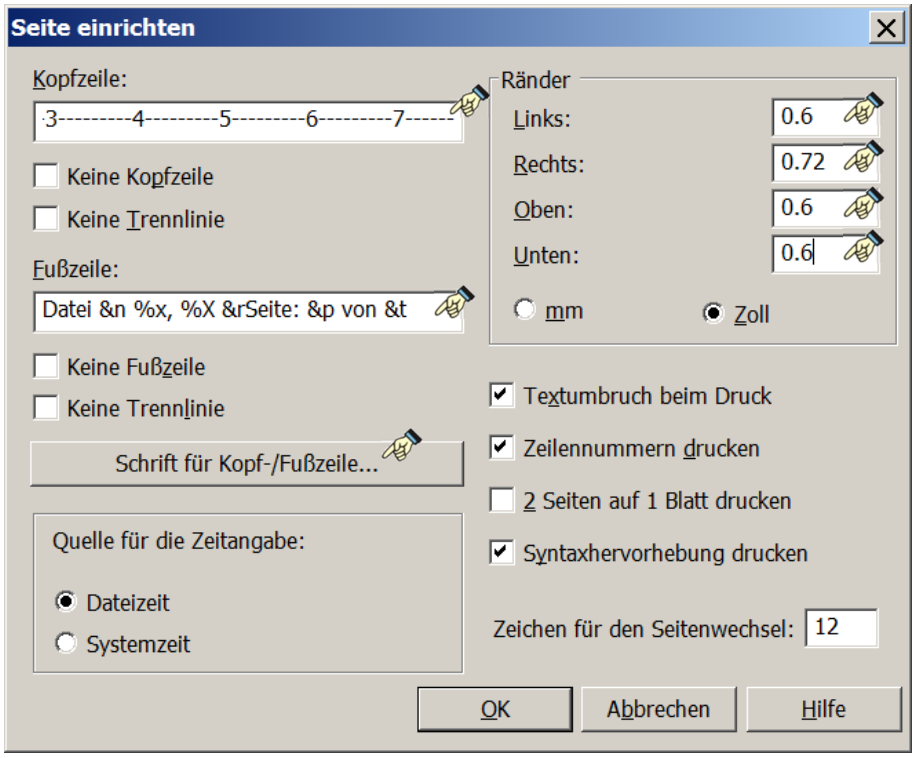


Bild 7.1.1-01: Seite einrichten im DIN A4 Hochformat

Das Hochformat wird eingestellt mit

Datei => Druckereinstellungen => Drucker einrichten... => Hochformat

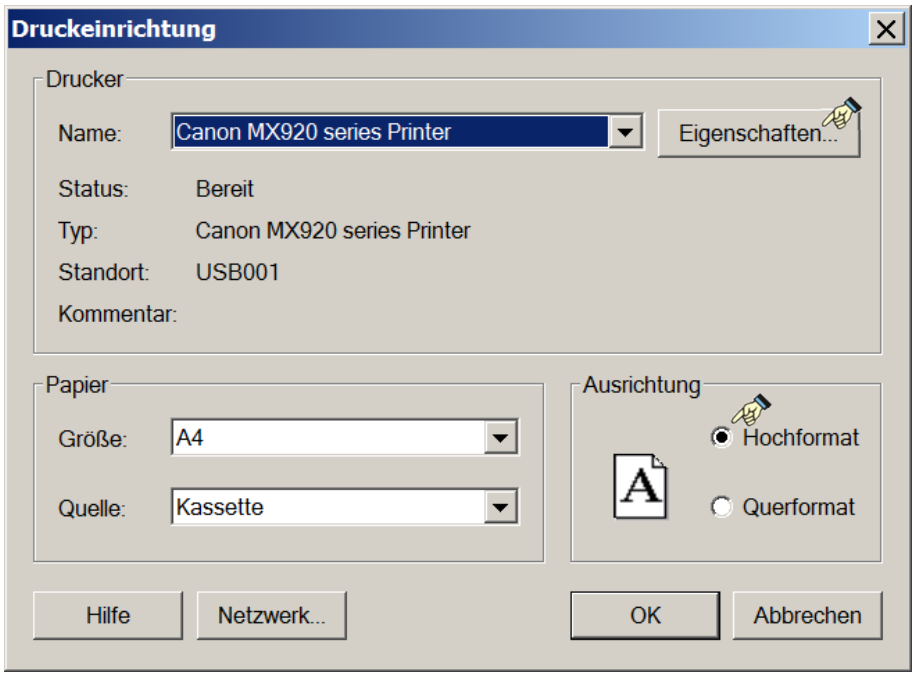


Bild 7.1.1-02: Drucker auf Hochformat einrichten

Ggf. müssen auch die individuellen Eigenschaften des Druckers eingestellt und angepasst werden! Bei manchen Druckern ist zusätzlich für häufig verwendete Formateinstellungen das Abspeichern und Wiederaufrufen - z.B. unter dem Namen **ASM-Source-Code** - derselben möglich.

Die 

wird auf **`Courier New - Standard - 10 pt`** eingestellt:

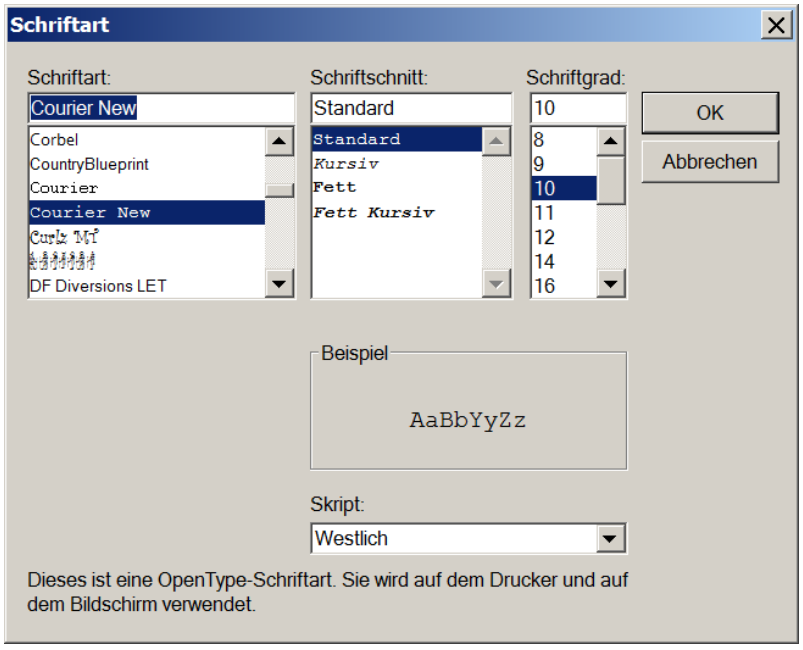


Bild 7.1.1-03: Schrift für Kopf- und Fußzeile auf `Courier New - Standard - 10 pt`

Damit der Text auch wirklich in **`Courier New - Standard - 10 pt`** dargestellt und gedruckt wird, muss auch noch unter

Ansicht => Schriftart einstellen... => &Schriftart

die Schriftart in gleicher Weise auf **`Courier New - Standard - 10 p`** eingestellt werden:

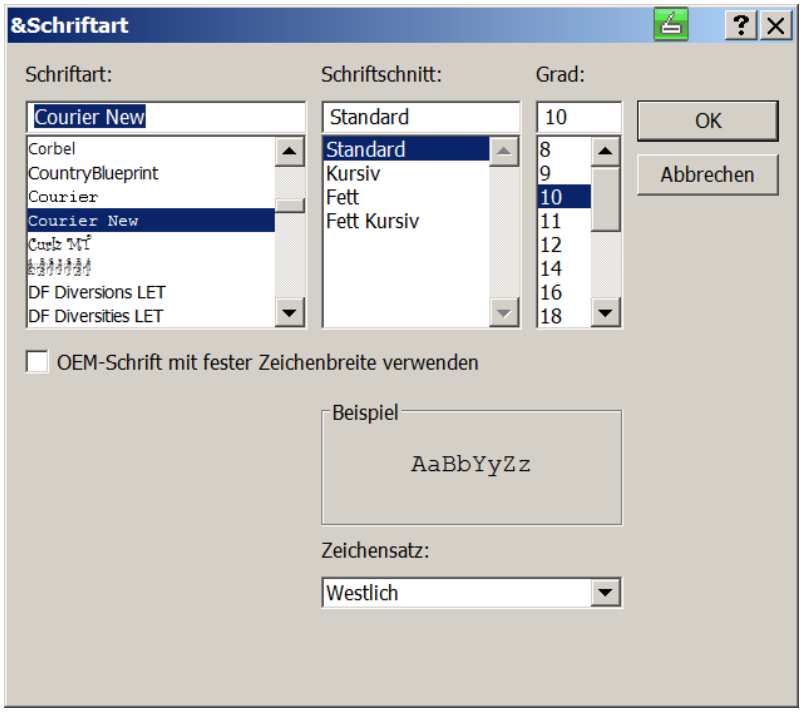
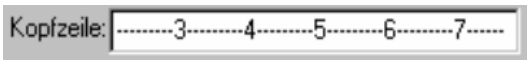


Bild 7.1.1-04: Schriften für Schriftart und Druckerschrift einstellen

Die Einstellung 

gibt nur einen Ausschnitt der folgenden vollständigen Eintragung wieder:

```
------|---------1---------2---------3---------4---------5---------6---------7------
```

Sie druckt als Kopfzeile die Spaltennummerierung mit vorlaufendem **------|** und eine Trennlinie.

Anmerkung: Man täusche sich ggf. nicht in der **Seitenansicht**, dass die Positionen der Text-Zeichen - insbesondere der letzten Zeichen in der Zeile - **nicht unbedingt identisch** mit den Positionen der Kopfzeile sind. Offensichtlich liegt das am Druckertreiber, da mit einem anderen Drucker dieses "Phänomen" nicht festgestellt wird. Der letzte Stern des Textes sollte mit dem letzten Bindestrich der Kopfzeile genau zusammentreffen:

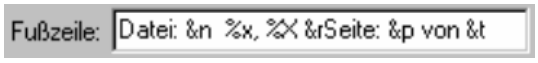
```
------|---------1---------2---------3---------4---------5---------6---------7------
    1 ;*****************************************************************************
    2 ; demo.asm      Syntaxhervorhebung fuer AVR-Assembler mit UltraEdit
    3 ;*****************************************************************************
    4 ;
    5 ; Hier werden die verschiedenen Syntaxhervorhebungen demonstriert.
    6 ; Im UltraEdit werden die Farben nach folgendem Schema bestimmt:
    7 ; Farbtiefe/Saettigung/Helligkeit:   fff/sss/hhh
```

Bild 7.1.1-05: Kopfzeile, Zeilennummerierung und nicht passender Beispieltext mit dem Drucker Canon MX920 Series

```
------|---------1---------2---------3---------4---------5---------6---------7------
    1 ;*****************************************************************************
    2 ; demo.asm      Syntaxhervorhebung fuer AVR-Assembler mit UltraEdit
    3 ;*****************************************************************************
    4 ;
    5 ; Hier werden die verschiedenen Syntaxhervorhebungen demonstriert.
    6 ; Im UltraEdit werden die Farben nach folgendem Schema bestimmt:
    7 ; Farbtiefe/Saettigung/Helligkeit:   fff/sss/hhh
```

Bild 7.1.1-06: Kopfzeile, Zeilennummerierung mit passendem Beispieltext, Drucker Epson ET-7700 Series

Im Ausdruck wird der Quell-Code jedoch - stets wie gewünscht - dargestellt!



Datei: &n %x, %X &rSeite: &p von &t

druckt als Fußzeile eine Trennlinie und darunter die verlangten Angaben:

```
   56 ; Schrift: Black, Fett
   57      ! ~ + - * = / % & > < ^ | ?
   58 ;
   59 ; C4"AVR-spezifische Namen"
   60 ; Farbtiefe/Saettigung/Helligkeit:   080/240/030
   61 ; Rot/Gruen/Blau:                    000/064/000
   62 ; Schrift: Olive, Fett
   63      ACSR, ADCH, DDRB, PORTB, PINB

Datei: 207 demo.asm 14.01.2020, 19:03:11                    Seite: 1 von 2
```

Bild 7.1.1-06: Fußzeile mit Angaben für Datei, Datum, Uhrzeit und Seitennummerierung

Anmerkung: Leider gibt es in der Anwendung UltraEdit keine Möglichkeit, das individuelle hier dargestellte Druckdesign zusammen mit der Datei abzuspeichern, so dass für jeden Wechsel der Syntaxhervorhebung auch wieder die Druckeinstellungen neu vorgenommen werden müssen.

Immerhin bleibt die Einstellung für das nächste Listing - sofern nur Assembler oder CVAVR verwendet wird - erhalten.

7.1.2 Einstellungen für C-Compiler-Programme

Die bevorzugte Einstellung für **C**-Programme bezieht sich auf das **DIN A4 Querformat**, so dass sich die Kopfzeile – natürlich ohne den hier dargestellten Zeilenumbruch – über die gesamte Seitenbreite erstreckt:

```
------|---------1---------2---------3---------4---------5---------6---------7------
---8---------9---------0---------1---------2
```

Damit wird der Kommentierung ein breiterer Raum zur Verfügung gestellt, es stehen in einer Zeile bis zu 120 Zeichen zur Verfügung. Die weiteren Einstellungen werden analog wie für Assembler-Programme vorgenommen.

Datei => Druckeinstellungen => Seite einrichten...

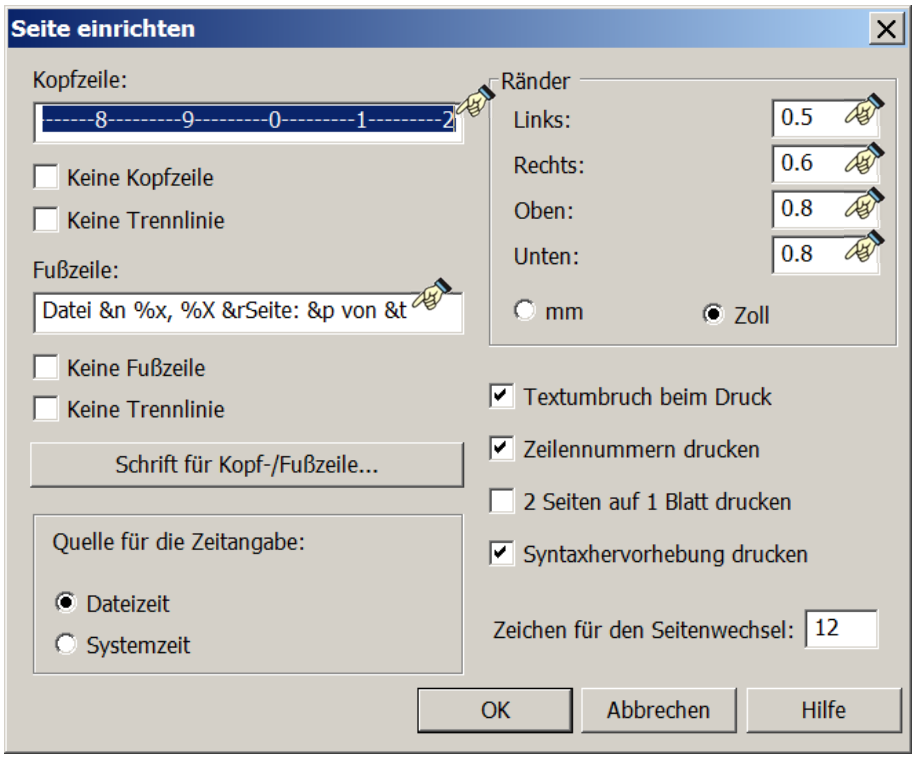
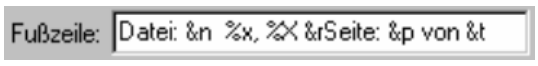


Bild 7.1.2-01: Seite einrichten im DIN A4 Querformat

Auch hier müssen alle Schriften auf **`Courier New - Standard - 10 pt`** eingestellt werden.

Die 

wird auf **`Courier New - Standard - 10 pt`** eingestellt.



`Datei: &n %x, %X &rSeite: &p von &t`

Datei => Druckereinstellungen => Drucker einrichten... => Querformat

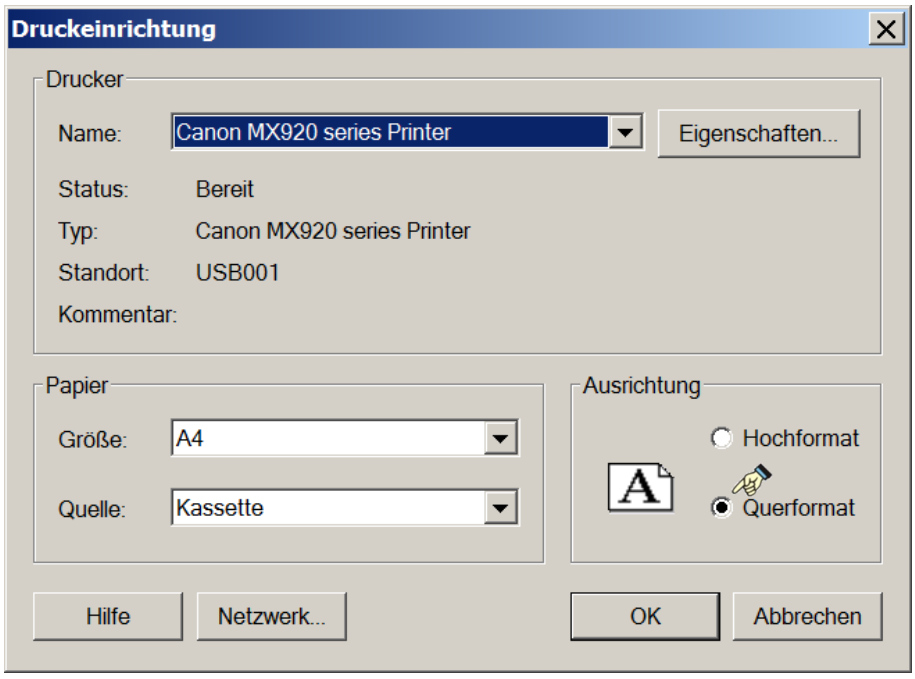


Bild 7.1.2-02: Canon-Drucker auf Querformat einrichten

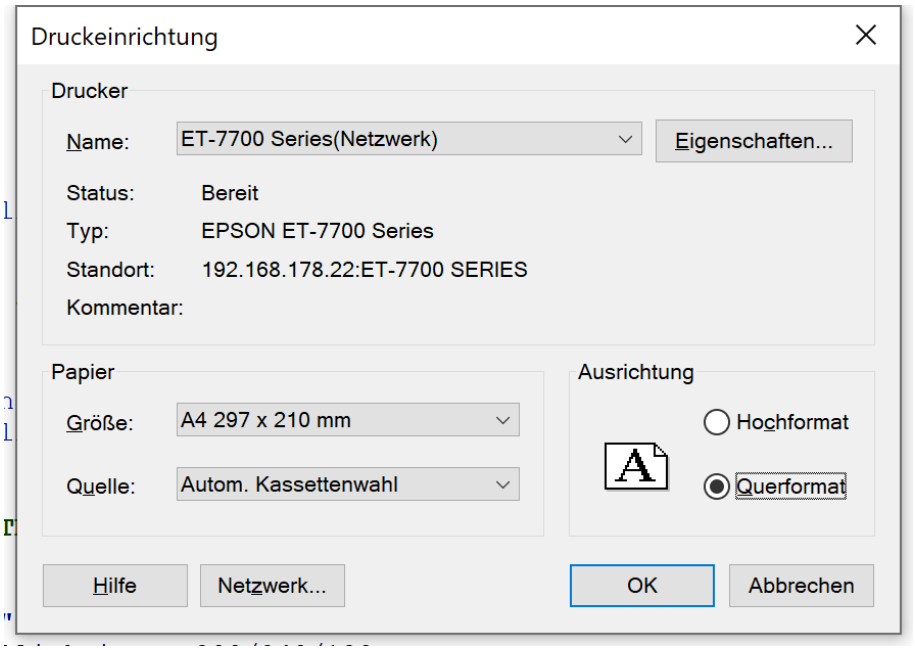


Bild 7.1.2-03: Epson-Drucker auf Querformat einrichten

7.2 Syntaxhervorhebung (Syntax Highlighting)

Hinter dem Begriff **Syntaxhervorhebung** bzw. **Syntax-Highlighting** verbirgt sich die Fähigkeit, vordefinierte Wörter zu erkennen und sie in unterschiedlichen Farben darzustellen. Dies ist insbesondere für verschiedene Programmier-Sprachen von großem Nutzen, kann sich aber auch für andere Anwendungen als nützlich erweisen, wenn bestimmte Wörter in einer Datei in einer anderen Farbe erscheinen sollen.

Wie bereits in der Einleitung angedeutet, kann man im Editor UltraEdit Quell-Codes zahlreicher Programmiersprachen in ihrer besonderen Syntax optisch durch Farben und Schriftstil kenntlich machen. Dazu ist für jede "Sprache" ("Language") eine eigene ***.uew-Datei** vorgesehen, deren interne Struktur mit sog. "Themen", die gesondert definiert werden, korrespondiert. Als Format ist die ***.uew-Datei** eine reine **Text-Datei** und kann auch als solche editiert werden. Die grundsätzliche Vorgehensweise, um UltraEdit für eine individuelle Editier-Hilfe und -Darstellung seiner "Sprache" zu nutzen, besteht deshalb aus zwei Schritten:

- **Strukturierung der Datei *.uew für die individuelle "Sprache" und**
- **Syntax in den "Themen" für die individuelle "Sprache" festlegen**

Ist die "Sprache" für die eigenen Zwecke nicht vorhanden, kann man sie - bevor man sie selbst erstellen muss - ggf. aus einer Sammlung verschiedenster "Languages", die UltraEdit zur Auswahl und zum Download in folgender Website anbietet, entnommen werden:

https://www.ultraedit.com/downloads/extras/wordfiles.html

Wir wollen einmal sehen, ob z.B. für den **AVR-Assembler** eine passende Datei angeboten wird. Und siehe da: es existiert eine **asmavr.uew** (Suche nach **AVR** mittels **Strg+F**):

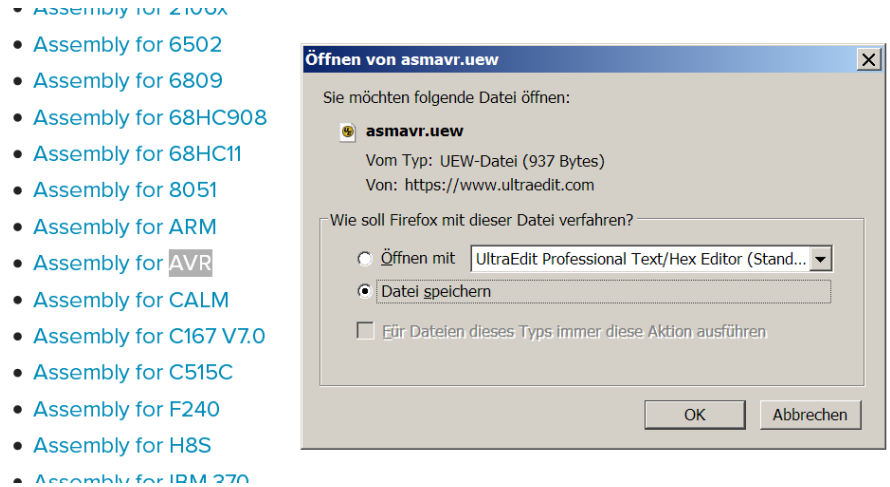


Bild 7.2-01: *.uew-Datei für den AVR-Assembler suchen und herunterladen

Sie wird heruntergeladen und im Systemordner abgelegt:

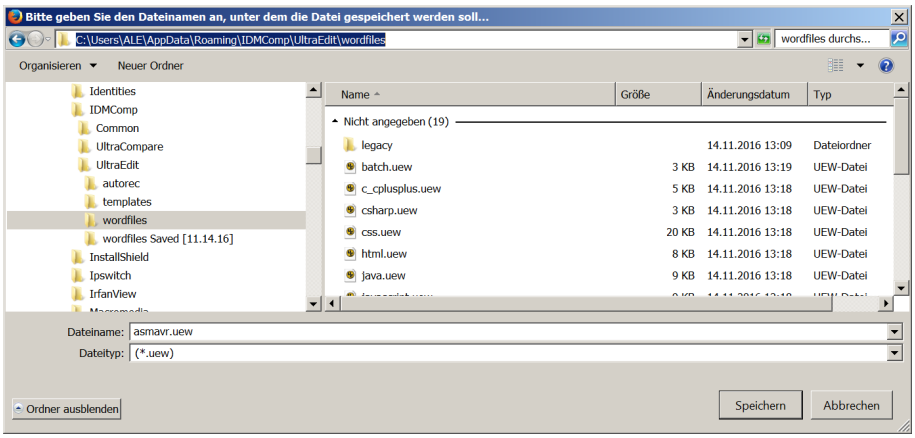


Bild 7.2-02: Gefundene Datei im Systemordner (s.o.) ablegen

Erst nach dem Neustart von UltraEdit ist die Datei **asmavr.uew** als "Highlighting Language" eingebunden und kann dann verwendet werden.

Nach dem Öffnen der Datei mit UltraEdit (Klick auf **Öffnen**) und erster Prüfung der relativ kleinen Datei **asmavr.uew** kann man feststellen, dass die Definitionen für die Assembler-Syntaxhervorhebung im Umfang jedoch erheblich reduziert sind. Es hat sich zwar an dem Aufbau der Syntax des Quell-Codes nichts geändert aber die definierten Wortgruppen sind doch sehr beschränkt.

So wie hier eine neue Datei eingefügt wurde, können die Dateien ***.uew** im Systemordner auch entfernt oder selbst erzeugte/ergänzte/geänderte hinzugefügt werden. Für die weitere Beschreibung ist es sinnvoll, die Datei **asmavr.uem** wieder zu löschen und die ursprüngliche Datei für den **AVR-Assembler** als **avr_asm.uew** nachzuladen. Desgleichen wird die neu aufgebaute **cvavr.uew** nachgeladen. Diese Dateien stehen ebenfalls zum Download bereit und sind unter **7.3 Wortsammlung für AVR-Assembler** und **7.4 Wortsammlung für CodeVisionAVR C-Compiler** in vollem Text wiedergegeben.

Extras => Optionen => Anzeige => (Schnellsuche:) Syntaxhervorhebung

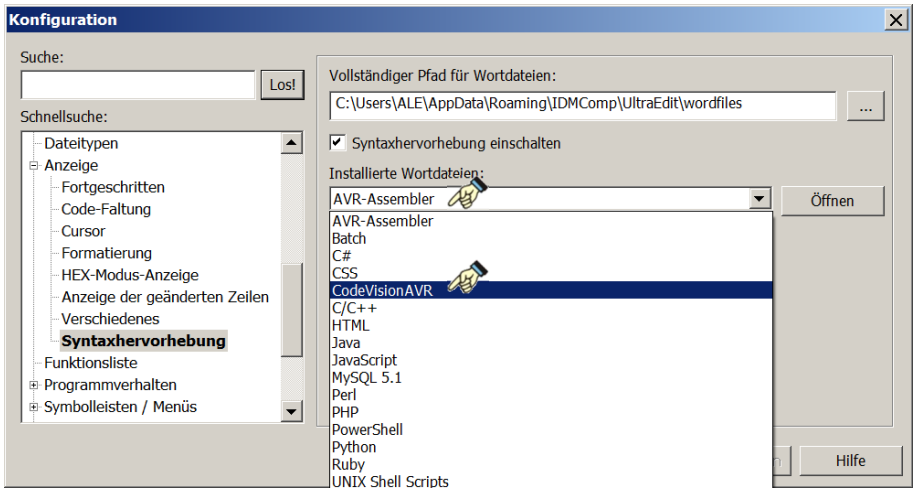


Bild 7.2-03: Syntaxhervorhebung für avr_asm.uew und cvavr.uew
(Man beachte, dass der Name von der Dateibezeichnung abweicht.)

7.2.1 AVR-Assembler

7.2.1.1 Syntaxbefehle für den AVR-Assembler

Jede Datei ***.uew** einer "Sprache" beginnt mit der Anweisung **`/Lx"Anwendungs-Text"`** wobei **`x`** für eine fortlaufende Zahl steht (hier: **`/L1"AVR-Assembler"`**). Zur weiteren Lektüre ist es sinnvoll, immer auch einen Blick auf den Quell-Code der **vollständigen Wortsammlung für den Assembler** zu werfen (über dem Link **`[„STRG und Klicken“]`** ist ein Hin- und Herspringen möglich).

1. Abschnitt von der Datei avr_asm.uew

```
/L1"AVR-Assembler" AASM_LANG Nocase Line Comment = ; Block Comment On = /*
Block Comment Off = */ String Chars = "' File Extensions = ASM INC
```

Darin bedeuten (in der Original-Anweisung erfolgt **kein** Zeilenumbruch):

`/L1`	Beginn der Wortsammlung des AVR-Assemblers (Anwendung **`1`**).
`"AVR-Assembler"`	Die Anwendung **`1`** wird mit **`AVR-Assembler`** benannt. Der Name der Sprache folgt direkt auf das **`/Lx`** und wird in Anführungszeichen gesetzt. Die Bezeichnung wird im Menü für die Syntax angezeigt (**Bild 7.2-03** und **Bild 7.2.1.2-01**). Die Bezeichnung darf bis zu 18 Zeichen lang sein.
`AASM_LANG`	Von UltraEdit vorgesehenes Schlüsselwort **`AASM_LANG`** für besondere Erweiterungen des Codings.
`Nocase`	Anwendung unterscheidet **nicht** zwischen Groß- und Kleinschreibung.
`Line Comment = ;`	Zeilenkommentare werden mit einem Semikolon **`;`** eingeleitet.
`Block Comment On = /*`	Blockkommentare werden mit **`/*`** eingeleitet.
`Block Comment Off = */`	Blockkommentare werden mit **`*/`** beendet.

`String Chars = "'`	Zeichenketten werden mit **"** **oder** **'** eingeschlossen.
`File Extensions = ASM INC`	Alle Dateien mit der Datei-Erweiterung **ASM** **oder** **INC** werden von UltraEdit mit der hier definierten Syntaxhervorhebung angezeigt und bearbeitet.

2. Abschnitt von der Datei avr_asm.uew - Wort-Begrenzungszeichen /Delimiters =

Um genau feststellen zu können, wo ein neues Wort beginnt und wo ein Wort endet, damit es mit der Wortsammlung einer gegebenen Sprache verglichen werden kann, müssen Trennzeichen oder Begrenzungszeichen definiert werden.

```
/Delimiters = ~!@%^&*()=|\/{}→[]:"'<> ,?/
```

Beachte: Auch ein **Leerzeichen** ist aufgeführt (vor dem Komma **,**), dagegen fehlt der Unterstrich, da er als "alphabetisches" Zeichen Bestandteil eines Wortes (Variable, Konstante usw.) sein kann.

3. Syntax-Abschnitte in der Datei avr_asm.uew für die zugeordneten Farbeinstellungen mit dem Gruppen-Index /Cn

Farbcodes für eine Gruppe lassen sich durch Einfügen einer Zeile mit **/Cn** am Zeilenanfang festlegen, wobei **n** der Wert des Farbindexes von **1** bis … ist. Unmittelbar nach dem **/Cn** folgt in Anführungszeichen die verbale Beschreibung bzw. Bezeichnung für die Gruppe. Diese wird im Menü für die Syntax (**Bild 7.2.1.2-01**) einer Sprache mit bis zu 24 Zeichen angezeigt.

Beispiel: Für die mnemotechnischen Codes der Assembler-Befehle soll eine einheitliche Farbe festgelegt werden. Die zugeordnete Gruppe **/C2** wird deshalb mit **"Instruktionen"** bezeichnet:

```
/C2"Instruktionen"
add adc adiw and andi asr
brbs brbc breq brne brcs brcc brsh brlo brmi brpl brge brlt brhs
brhc brts brtc brvs brvc brie brid bst bld bset bclr break
cbr clr com call cpse cp cpc cpi cbi clc cln clz cli cls clv clt clh
dec
eor eicall eijmp elpm
fmul fmuls fmulsu
inc ijmp icall in
jmp
ldi lds ld ldd lpm lsl lsr
mul muls mulsu mov movw
neg nop
or ori out
push pop
rjmp rcall ret reti ror rol
sub subi sbc sbci sbiw sbr ser sbrc sbrs sbic sbis st sts std spm
sbi sec sen sez sei ses sev set seh swap sleep
tst
wdr
```

Beachte: In einer Zeile dürfen nur Begriffe stehen, die mit demselben Zeichen beginnen. Begriffe mit demselben Anfangszeichen dürfen sich auch über mehrere Zeilen erstrecken.

Insgesamt sind für den AVR-Assembler 8 Gruppen (Farb-Abschnitte **/C1**, **/C2** bis **/C8**) definiert worden (siehe auch **Bild 7.2.1.2-01**):

```
/C1"Hexadezimal-Zahlen"
/C2"Instruktionen"
/C3"Operatoren/Symbole"
/C4"AVR-spezifische Namen"
/C5"Arbeits-Register"
/C6"Funktionen"
/C7"Preprozessor-Anweisungen"
/C8"Direktiven"
```

7.2.1.2 Farben und Schriftstile der Gruppen für den AVR-Assembler

In jeder Sammlung von Wörtern einer "Sprache" können verschiedene Wortgruppen definiert werden, die dann verschiedene Farben und Schriftstile erhalten. Für den AVR-Assembler sind 8 solcher Wortgruppen definiert worden, denen im Folgenden Farben und Schriftstile zugeordnet werden sollen.

Die Einstellungen werden im Fenster **Themen verwalten** vorgenommen.

Ansicht => Themen => Themen verwalten... => Syntax => Sprache: => AVR-Assembler

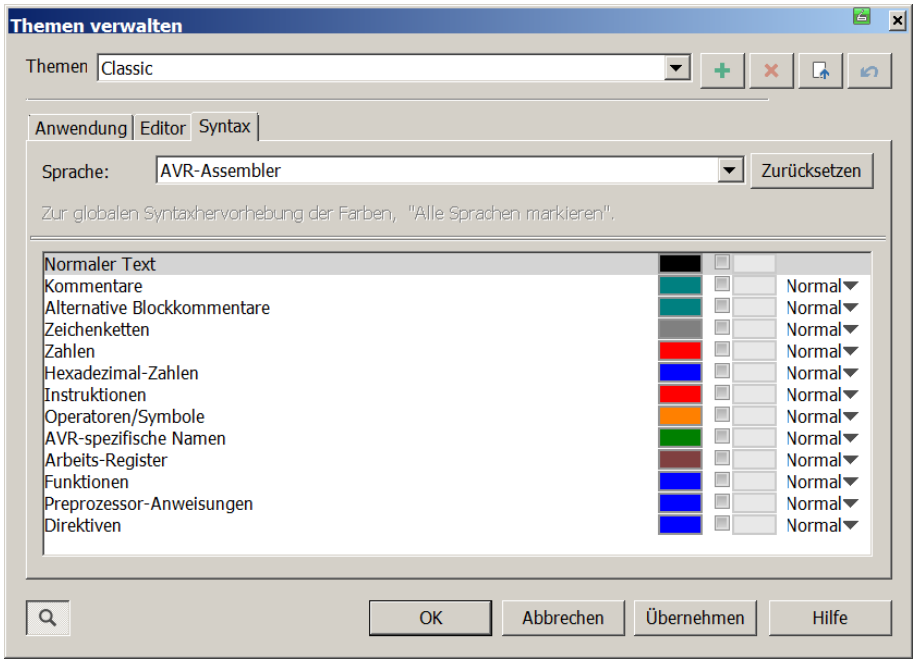


Bild 7.2.1.2-01: Ursprungs-Menü der Syntax für den AVR-Assembler

Im **Bild 7.2.1.2-01** werden die Wortgruppen im Fenster der Wortsammlung **AVR-Assembler** angezeigt:

- Vordefiniert: **Normaler Text** (Wörter, die nicht erkannt werden)
- Vordefiniert: **Kommentare** (Wörter einer Kommentarzeile)
- Vordefiniert: **Alternative Blockkommentare**
- Vordefiniert: **Zeichenketten**
- Vordefiniert: **Zahlen** (Wörter, die mit einer Ziffer [0-9] beginnen)
- Individuell definiert durch **/Cn** (**n** von **1** bis ...) für Gruppen von zusammengehörigen Begriffen
  - **/C1** **Hexadezimal-Zahlen**
  - **/C2** **Instruktionen**
  - **/C3** **Operatoren/Symbole**
  - **/C4** **AVR-spezifische Namen**
  - **/C5** **Arbeits-Register**
  - **/C6** **Funktionen**
  - **/C7** **Preprozessor-Anweisungen**
  - **/C8** **Direktiven**

Als Beispiel soll die Farbe und der Schriftstil für die Wortgruppe **/C2"Instruktionen"** des AVR-Assemblers vorgenommen werden: **Schwarze** Zeichen in **Fett**schrift. Für alle weiteren Wortgruppen ist die Prozedur die gleiche. Am Ende des Abschnitts gibt eine Zusammenfassung alle Einstellungen wieder.

Die Wortgruppen **Normaler Text**, **Kommentare**, **Alternative Blockkommentare**, **Zeichenketten** und **Zahlen** sind vordefiniert, können aber in gleicher Weise hier ebenfalls angepasst werden.

Ansicht => Themen => Themen verwalten... => Syntax => Sprache: => AVR-Assembler

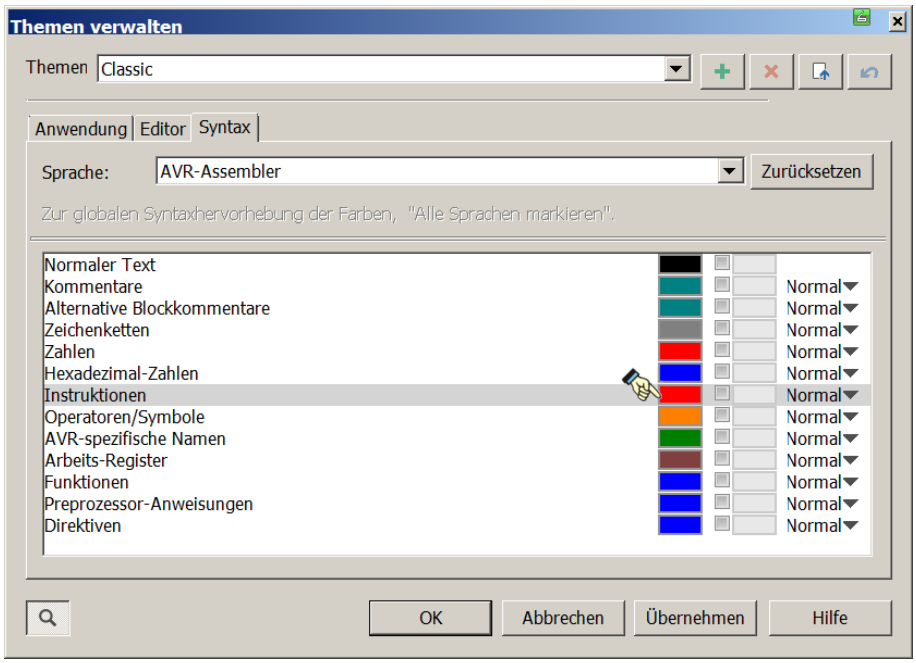


Bild 7.2.1.2-02: Menü, um die Syntax für den AVR-Assembler festzulegen

Ein Klick auf das **Farbfeld** der Zeile **Instruktionen** und es stehen zahlreiche Farben mit dem Menü für die Farbeinstellung zur Verfügung:

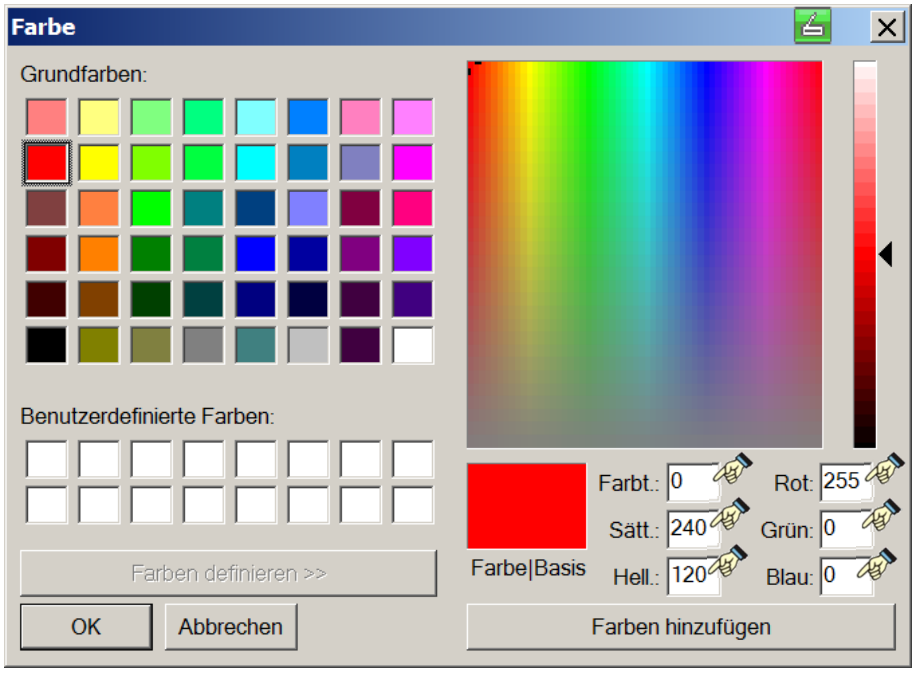


Bild 7.2.1.2-03: Menü, um Farbe für die Wortgruppe "Instruktionen" festzulegen

Die Farbauswahl wird gewöhnlich aus dem Bereich der Grundfarben ausgewählt. Zur eindeutigen Definition werden, da vom Editor keine Farbnamen genannt werden, folgende Angaben gemacht:

Farbtiefe/Sättigung/Helligkeit sowie **Rot/Grün/Blau**

Für die **Instruktionen** soll die Einstellung auf **160/000/000** und **000/000/000** (Farbe **Schwarz**) vorgenommen werden:

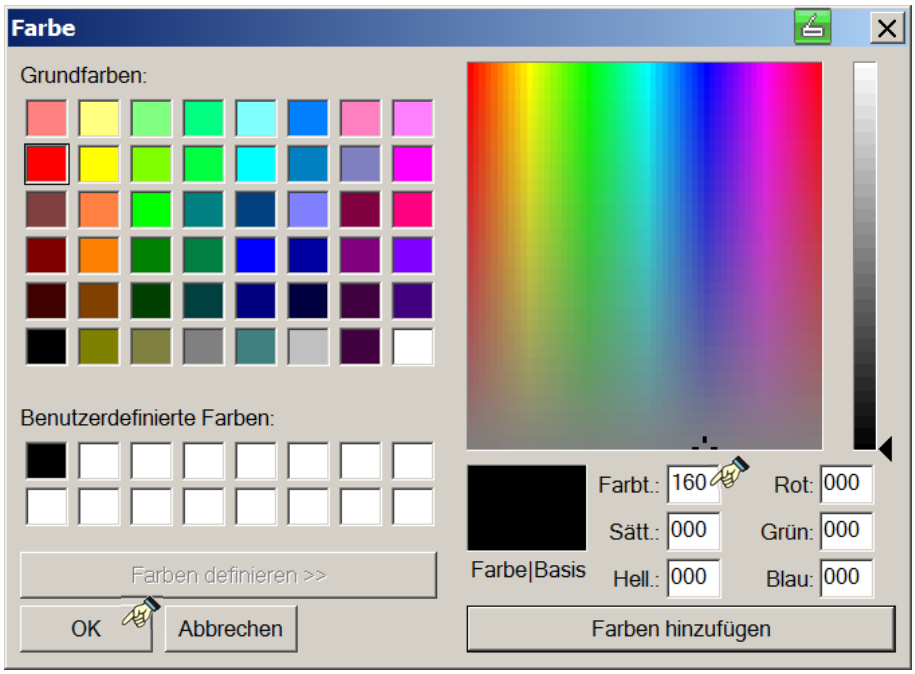


Bild 7.2.1.2-04: Wortgruppe "Instruktionen" wird festgelegt auf 160/000/000 und 000/000/000

Rechts von jedem Vordergrund-Farbfeld befindet sich ein kleines quadratisches Feld **Hintergrundfarbe aktivieren**. Wenn diese Option markiert wird, kann für das markierte Element eine Hintergrundfarbe verwendet werden. Dieses Kästchen bleibt deaktiviert.

Rechts vom Hintergrund-Farbfeld befindet sich das Feld **Schriftstile**. Es ist zunächst mit **Normal** vorbesetzt und kann mit dem Schriftstil **Normal, Fett, Kursiv** oder **Unterstreichen** angegeben werden. Für die Gruppe der Instruktionen soll **Fett** gewählt werden:

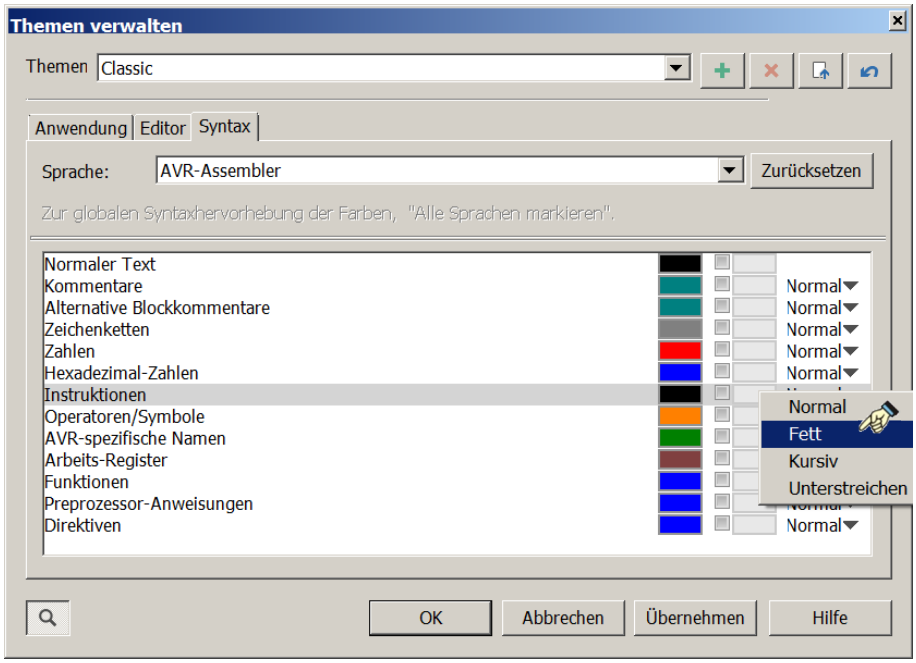


Bild 7.2.1.2-05: Wortgruppe "Instruktionen" wird festgelegt auf den Schriftstil "Fett"

Auf die gleiche Weise werden für alle Wortgruppen die Einstellungen vorgenommen, wie sie in der folgenden Zusammenstellung definiert sind.

Die ersten fünf Farbgruppen werden von UltraEdit automatisch bereitgestellt. Auch deren Farbe, Schriftstil und Hintergrundfarbe kann individuell eingestellt werden. Der Schriftstil ist ausschließlich **`Courier New, Font Size 10`**.

- **Normaler Text** `Black, Normal` **160/000/000** und **000/000/000**
- **Kommentare** `Navy, Normal` **160/240/060** und **000/000/128**
- **Alternative Blockkommentare** `Waldgrün, Normal` **100/240/048** und **000/102/051**
- **Zeichenketten** **`Blau, Fett`** **160/240/120** und **000/000/255**
- **Zahlen** **`Rot, Fett`** **000/240/120** und **255/000/000**

Die alternativen Blockkommentare werden in derselben Farbe dargestellt, wie die sonstigen Kommentare, obgleich ein anderer Farb-Code gewählt wurde. Die weiteren acht Farbgruppen sind mit **`/C1`** bis **`/C8`** definiert worden. Deren Farbe, Schriftstil und Hintergrundfarbe wird ebenso individuell eingestellt:

- **`/C1"Hexadezimal-Zahlen"`** **`Rot, Fett`** **000/240/120** und **255/000/000**
- **`/C2"Instruktionen"`** **`Black, Fett`** **160/000/000** und **000/000/000**
- **`/C3"Operatoren/Symbole"`** **`Black, Fett`** **160/000/000** und **000/000/000**
- **`/C4"AVR-spezifische Namen"`** **`Oliv, Fett`** **080/240/030** und **000/064/000**
- **`/C5"Arbeits-Register"`** **`Indigo, Fett`** **180/240/120** und **128/000/255**
- **`/C6"Funktionen"`** **`Violett, Fett`** **180/240/046** und **049/000/098**
- **`/C7"Preprozessor-Anweisungen"`** **`Grün, Fett`** **080/240/060** und **000/128/000**
- **`/C8"Direktiven"`** **`Dunkelrot, Fett`** **000/240/060** und **128/000/000**

Und so sieht die Einstellung zum Schluss aus:

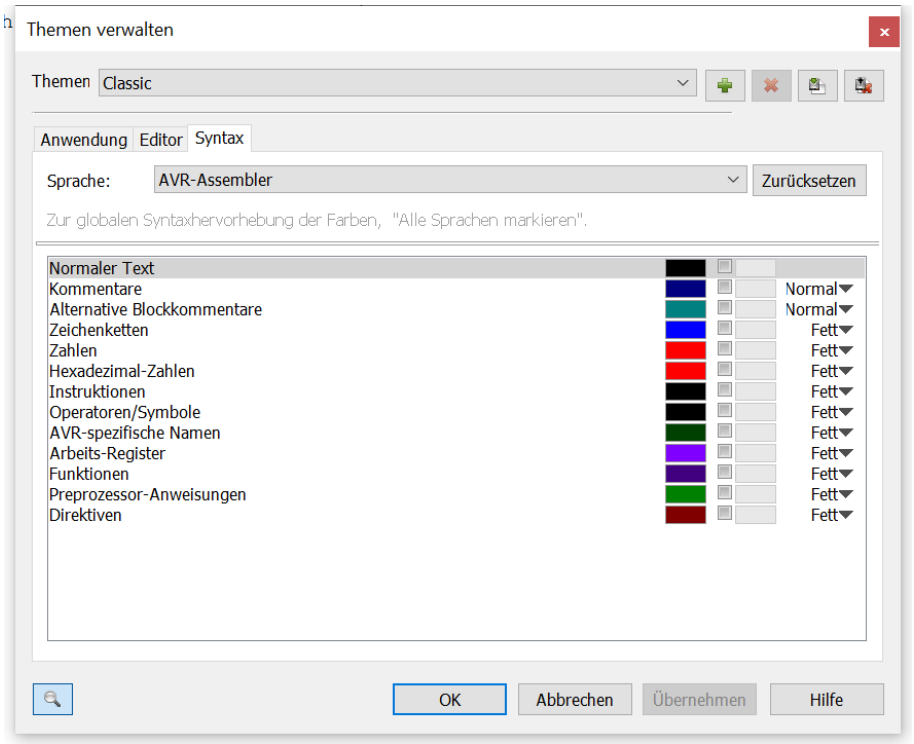


Bild 7.2.1.2-06: Alle Wortgruppen für den AVR-Assembler sind festgelegt

Die Hintergrundfarbe wird stets auf **weiß** belassen. Es ist natürlich Geschmacksache, für jede Farbgruppe eine individuelle Farbe und/oder einen besonderen Schriftstil einzustellen. Aber "Allzuviel ist ungesund"!

7.2.2 CodeVision AVR

7.2.2.1 Syntaxbefehle für den CodeVision AVR C-Compiler

Wie für den AVR-Assembler werden für den CodeVisionAVR C-Compiler spezielle Wortgruppen definiert. Zur weiteren Lektüre ist es sinnvoll, auch hier immer einen Blick auf den Quell-Code der **vollständigen Wortsammlung für den CodeVisionAVR C-Compiler** zu werfen (über dem Link **[„STRG und Klicken“]** ist ein Hin- und Herspringen möglich).

1. Abschnitt von der CVAVR-Wortsammlung in der Datei cvavr.uew:

```
/L2"CodeVisionAVR" Line Comment = // Block Comment On = /* Block Comment
Off = */ Escape Char = \ String Chars = "' File Extensions = C CPP CC CXX H
HPP AWK GES i lib
```

Darin bedeuten (in der Original-Anweisung erfolgt **kein** Zeilenumbruch):

`/L2`	Beginn der Wortsammlung des C-Compilers CVAVR (Anwendung **2**).
`"CodeVisionAVR"`	Die Anwendung **2** wird mit **CodeVisionAVR** benannt.
`Line Comment = //`	Zeilenkommentare werden mit **//** eingeleitet.
`Block Comment On = /*`	Blockkommentare werden mit **/*** eingeleitet.
`Block Comment Off = */`	Blockkommentare werden mit ***/** beendet.
`Escape Char = \`	Als Escape-Zeichen wird der Backslash **** verwendet.
`String Chars = "'`	Zeichenketten werden mit **"** **oder** **'** eingeschlossen.
`File Extensions = C CPP CC CXX H HPP AWK GES i lib`	Alle Dateien mit der Datei-Erweiterung **C**, **CPP**, **CC**, **CXX**, **H**, **HPP**, **AWK**, **GES**, **i** und **lib** werden von UltraEdit mit der hier definierten Syntaxhervorhebung angezeigt. **GES** ist als Datei-Erweiterung ergänzt worden, um alle Quell-Dateien eines Projektes in einer Datei mit dieser Erweiterung zusammenzuführen. So ist man in der Lage, das gesamte Projekt mit gleicher Syntax-Hervorhebung zu untersuchen und Zusammenhänge besser darstellen zu können.

2. Abschnitt von der CVAVR-Wortsammlung - Wort-Begrenzungszeichen `/Delimiters =`

Zur Definition der Trennzeichen wird eine NEUE Zeile in die Wortdatei eingefügt werden, die folgendes Aussehen erhält:

```
/Delimiters = ~!@%^&*()-+=|\/{}[]:;"'<> ,→.?
```

Dabei ist zu beachten, dass das Leerzeichen und das Tabulatorzeichen zu den Trennzeichen gehören und diese als Begrenzungen berücksichtigt werden sollen. Dagegen fehlt der Unterstrich, da er als "alphabetisches" Zeichen Bestandteil eines Wortes (Variable, Konstante usw.) sein kann. Die Zeile muss mit **/Delimiters =** beginnen.

3. Syntax-Abschnitte von der Datei cvavr.uew für die zugeordneten Farbeinstellungen mit dem Gruppen-Index `/Cn`

Farbcodes für eine Gruppe lassen sich durch Einfügen einer Zeile mit **/Cn** am Zeilenanfang festlegen, wobei **n** der Wert des Farbindexes von **1** bis … ist. Unmittelbar nach dem **/Cn** folgt in Anführungszeichen die verbale Beschreibung bzw. Bezeichnung für die Wortgruppe. Diese wird im Menü für die Syntax der Sprache mit bis zu 24 Zeichen angezeigt.

Beispiel: Für alle Schlüsselwörter des CodeVisionAVR C-Compilers soll eine einheitliche Farbe festgelegt werden. Die Wortgruppe **/C1** wird deshalb mit **"Schluesselwoerter"** bezeichnet:

```
/C1"Schluesselwoerter"
auto
break bit bool
_Bool __interrupt
case char const continue
default do double defined
else enum extern
float for
```

```
goto
if inline int interrupt
long
register return
short signed sizeof sfrb sfrw static struct switch
typedef
union unsigned
void volatile
while
```

Beachte: In einer Zeile dürfen nur Begriffe stehen, die mit demselben Zeichen beginnen. Begriffe mit demselben Anfangszeichen dürfen sich auch über mehrere Zeilen erstrecken. Beim Compiler CVAVR werden Groß- und Kleinbuchstaben unterschieden und als unterschiedliche Zeichen interpretiert.

Neben **/C1** sind für den CVAVR C-Compiler noch 7 weitere Farb-Abschnitte (**/C2** bis **/C8**) definiert worden (siehe auch **Bild 7.2.2.2-06**):

```
/C2"Symbole"
/C3"Operatoren"
/C4"AVR-spezifische Namen"
/C5"Speicherdefinitionen"
/C6"Typ-Synonyme"
/C7"Preprozessor-Anweisungen"
/C8"C++ Keywords"
```

7.2.2.2 Farben und Schriftstile der Gruppen für den CVAVR C-Compiler

In jeder Sammlung von Wörtern einer "Sprache" können verschiedene Wortgruppen definiert werden, die dann verschiedene Farben und Schriftstile erhalten. Für den CVAVR C-Compiler sind 7 solcher Wortgruppen definiert worden, denen im Folgenden Farben und Schriftstile zugeordnet werden sollen.

Die Einstellungen werden im Fenster **Themen verwalten** vorgenommen.

Ansicht => Themen => Themen verwalten... => Syntax => Sprache: => CodeVisionAVR

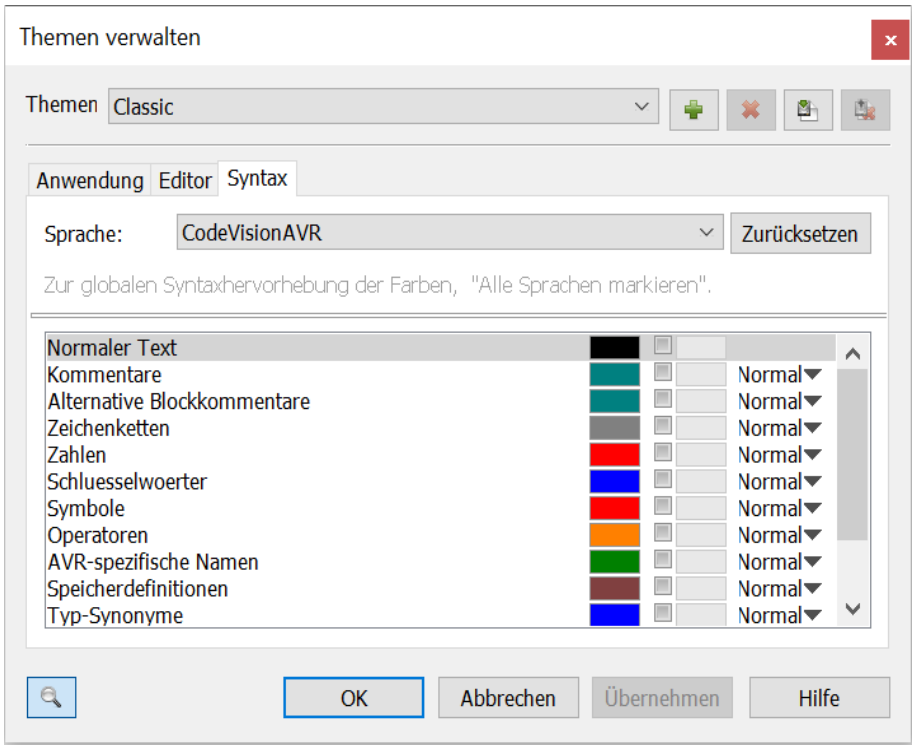


Bild 7.2.2.2-01: Ursprungs-Menü der Syntax für den CVAVR C-Compiler

Im **Bild 7.2.2.2-01** werden die Wortgruppen im Fenster der Wortsammlung **AVR-Assembler** zum Teil angezeigt (durch Skrollen oder Vergrößerung des Fensters werden auch **/C7** und **/C8** angezeigt).

- Vordefiniert: **Normaler Text** (Wörter, die nicht erkannt werden)
- Vordefiniert: **Kommentare** (Wörter einer Kommentarzeile)
- Vordefiniert: **Alternative Blockkommentare**
- Vordefiniert: **Zeichenketten**
- Vordefiniert: **Zahlen** (Wörter, die mit einer Ziffer [0-9] beginnen)
- Individuell definiert durch **/Cn** (**n** von **1** bis ...) für Gruppen von zusammengehörigen Begriffen
  - **/C1** **Schluesselwoerter**
  - **/C2** **Symbole**
  - **/C3** **Operatoren**
  - **/C4** **AVR-spezifische Namen**
  - **/C5** **Speicherdefinitionen**
  - **/C6** **Typ-Synonyme**
  - **/C7** **Preprozessor-Anweisungen**
  - **/C8** **C++ Keywords**

Als Beispiel soll die Farbe und der Schriftstil für die Wortgruppe **/C1"Schluesselwoerter"** des C-Compilers vorgenommen werden: **Schwarze** Zeichen in **Fett**schrift. Für alle weiteren Wortgruppen ist die Prozedur die gleiche, so dass für die übrigen Einstellungen auf das Ende des Abschnitts verwiesen wird.

Die Wortgruppen **Normaler Text**, **Kommentare**, **Alternative Blockkommentare**, **Zeichenketten** und **Zahlen** sind vordefiniert, können aber in gleicher Weise hier angepasst werden.

Ansicht => Themen => Themen verwalten... => Syntax => Sprache: => AVR-Assembler

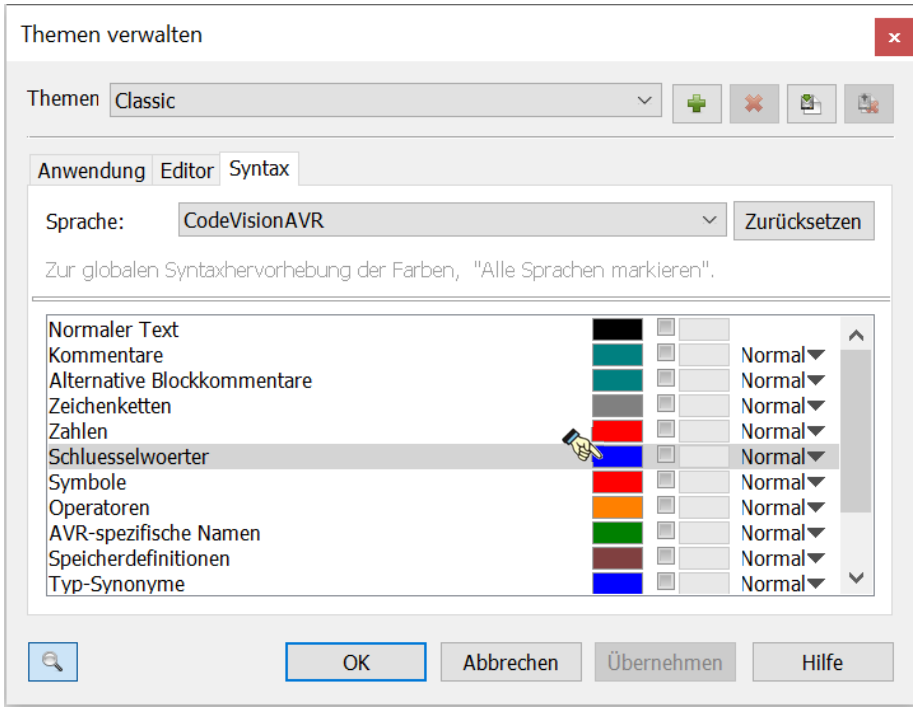


Bild 7.2.2.2-02: Menü, um die Syntax für den CVAVR C-Compiler festzulegen

Ein Klick auf das **Farbfeld** der Zeile **Schluesselwoerter** und es stehen zahlreiche Farben mit dem Menü für die Farbeinstellung zur Verfügung:

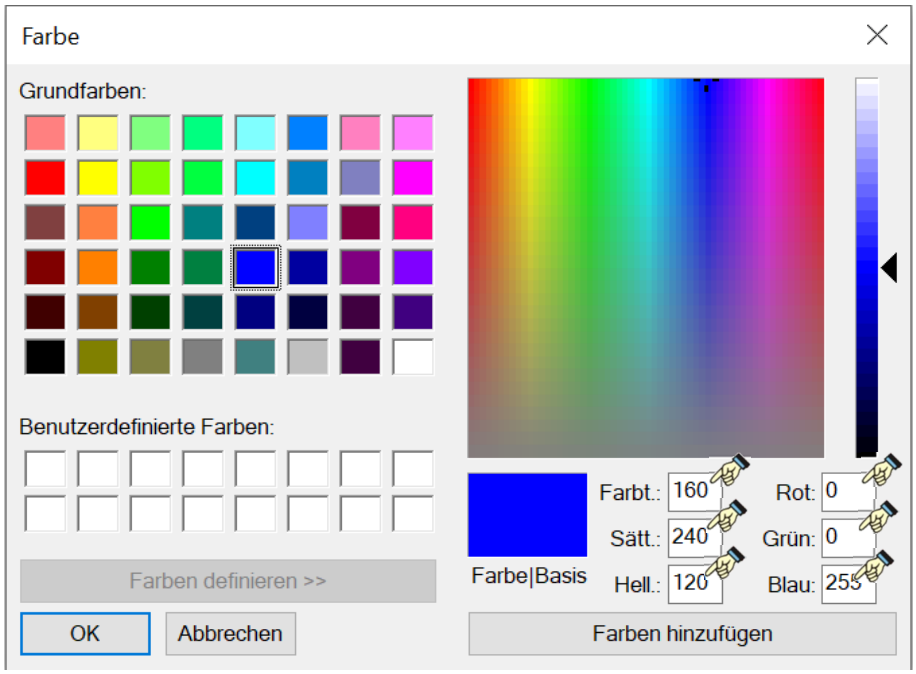


Bild 7.2.2.2-03: Menü, um die Farbe für die Schluesselwoerter festzulegen

Die Farbauswahl wird gewöhnlich aus dem Bereich der Grundfarben ausgewählt. Zur eindeutigen Definition werden, da vom Editor keine Farbnamen genannt werden, folgende Angaben gemacht:

Farbtiefe/Sättigung/Helligkeit sowie **Rot/Grün/Blau**

Für die **Schluesselwoerter** soll die Einstellung auf **160/000/000** und **000/000/000** (Farbe **Schwarz**) vorgenommen werden:

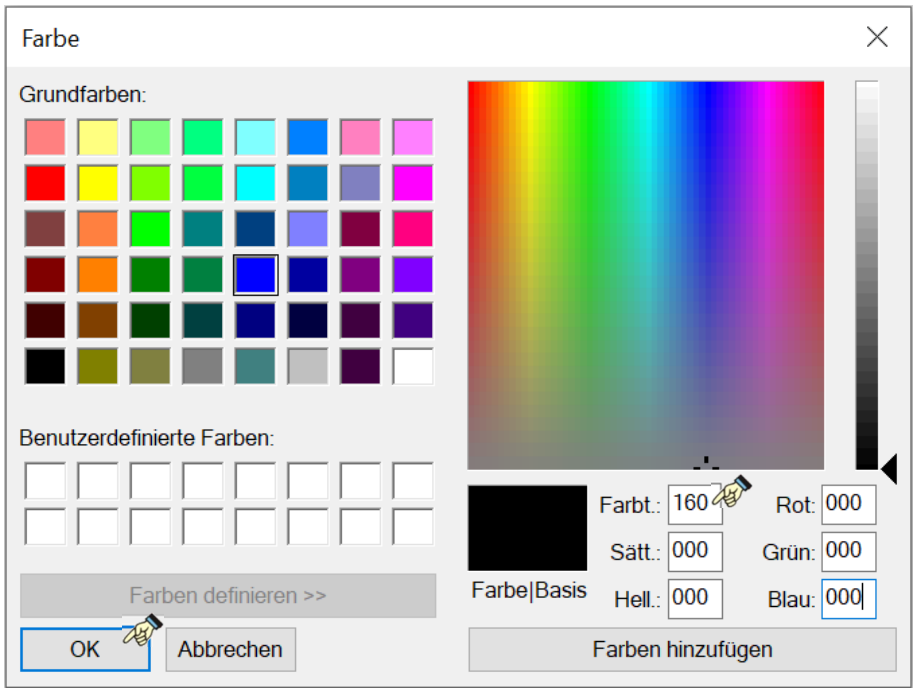


Bild 7.2.2.2-04: Wortgruppe "Instruktionen" festgelegt auf 160/000/000 und 000/000/000

Rechts neben dem Vordergrund-Farbfeld befindet sich ein kleines quadratisches Feld **Hintergrundfarbe aktivieren**. Wenn diese Option markiert wird, kann für das markierte Element eine Hintergrundfarbe verwendet werden. Dieses Kästchen bleibt deaktiviert.

Rechts vom Hintergrund-Farbfeld befindet sich das Feld **Schriftstile**. Es ist zunächst mit **Normal** vorbesetzt und kann mit dem Schriftstil **Normal, Fett, Kursiv** oder **Unterstreichen** angegeben werden. Für die Gruppe der Instruktionen soll **Fett** gewählt werden:

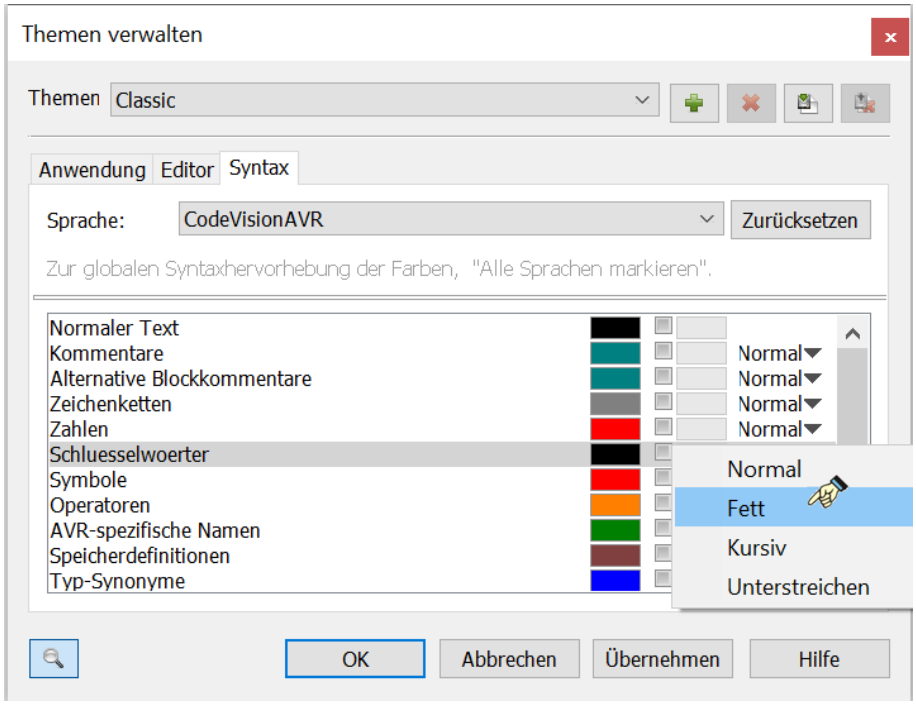


Bild 7.2.2.2-05: Wortgruppe "Schluesselwoerter" wird festgelegt auf Schriftstil "Fett"

Auf die gleiche Weise werden für alle Wortgruppen die Einstellungen vorgenommen, wie sie in der folgenden Zusammenstellung definiert sind.

Die ersten fünf Farbgruppen werden auch für C-Programme von UltraEdit automatisch bereitgestellt. Auch deren Farbe, Schriftstil und Hintergrundfarbe kann individuell eingestellt werden. Der Schriftstil ist ausschließlich **`Courier New, Font Size 10`**.

- **Normaler Text** `Black, Normal` **160/000/000** und **000/000/000**
- **Kommentare** `Navy, Normal` **160/240/060** und **000/000/128**
- **Alternative Blockkommentare** `Waldgrün, Normal` **100/240/048** und **000/102/051**
- **Zeichenketten** **`Blau, Fett`** **160/240/120** und **000/000/255**
- **Zahlen** **`Rot, Fett`** **000/240/120** und **255/000/000**

Die weiteren acht Farbgruppen sind mit **`/C1`** bis **`/C8`** definiert worden. Deren Farbe, Schriftstil und Hintergrundfarbe wird ebenso individuell eingestellt:

- **`/C1"Schluesselwoerter"`** **`Black, Fett`** **160/000/000** und **000/000/000**
- **`/C2"Symbole"`** **`Black, Fett`** **160/000/000** und **000/000/000**
- **`/C3"Operatoren"`** **`Black, Fett`** **160/000/000** und **000/000/000**
- **`/C4"AVR-spezifische Namen"`** **`Olive, Fett`** **080/240/030** und **000/064/000**
- **`/C5"Speicherdefinitionen"`** **`Fuchsia, Fett`** **180/240/120** und **128/000/255**
- **`/C6"Typ-Synonyme"`** **`Dunkelrot, Fett`** **000/240/060** und **128/000/000**
- **`/C7"Preprozessor-Anweisungen"`** **`Grün, Fett`** **080/240/060** und **000/128/000**
- **`/C8"C++ Keywords"`** **`Orange, Fett`** **020/240/120** und **255/128/000**

Und so sieht die Einstellung zum Schluss aus:

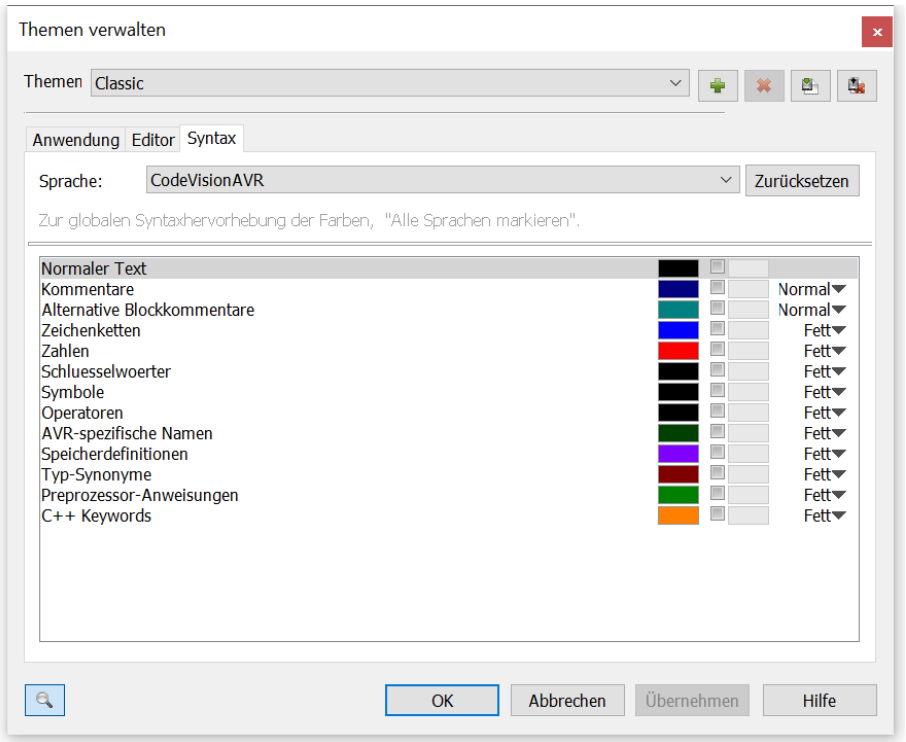


Bild 7.2.2.2-06: Alle Wortgruppen für den CodeVisionAVR C-Compiler sind festgelegt

7.3 Wortsammlung für AVR-Assembler

Die vollständige Wortsammlung für den Assembler in der Datei avr_asm.uew

```
/L1"AVR-Assembler" AASM_LANG Nocase Line Comment = ; Block Comment On = /*
Block Comment Off = */ String Chars = "' File Extensions = ASM INC

/Delimiters = ~!@%^&*()=|\/{}6[]:"'<> ,?/

/C1"Hexadezimal-Zahlen"
** $

/C2"Instruktionen"
add adc adiw and andi asr
brbs brbc breq brne brcs brcc brsh brlo brmi brpl brge brlt brhs
brhc brts brtc brvs brvc brie brid bst bld bset bclr break
cbr clr com call cpse cp cpc cpi cbi clc cln clz cli cls clv clt clh
dec
eor eicall eijmp elpm
fmul fmuls fmulsu
inc ijmp icall in
jmp
ldi lds ld ldd lpm lsl lsr
mul muls mulsu mov movw
neg nop
or ori out
push pop
rjmp rcall ret reti ror rol
sub subi sbc sbci sbiw sbr ser sbrc sbrs sbic sbis st sts std spm
sbi sec sen sez sei ses sev set seh swap sleep
tst
wdr
```

```
/C3"Operatoren/Symbole"
!
~
+
-
*
=
// /
%
&
>
<
^
|
?

/C4"AVR-spezifische Namen"
ACBG
ACD
ACI
ACIC
ACIE
ACIS0 ACIS1
ACME
ACO
ACSR
ADATE
ADC_INT
ADCCaddr ACIaddr
ADC0D ADC1D ADC2D ADC3D ADC4D ADC5D
ADCH ADCH0 ADCH1 ADCH2 ADCH3 ADCH4 ADCH5 ADCH6 ADCH7
ADCL ADCL0 ADCL1 ADCL2 ADCL3 ADCL4 ADCL5 ADCL6 ADCL7
ADCSRA ADCSRB
ADCW
ADEN
ADIE
ADIF
ADLAR
ADMUX
ADPS0 ADPS1 ADPS2
ADSC
ADTS0 ADTS1 ADTS2
AIN0D AIN1D
ANA_COMP
AS2
ASSR
BLBSET BLB01 BLB02 BLB11 BLB12
BODLEVEL0 BODLEVEL1 BODLEVEL2
BODS BODSE BORF
BOOTRST BOOTSZ0 BOOTSZ1
CAL0 CAL1 CAL2 CAL3 CAL4 CAL5 CAL6 CAL7
CKSEL0 CKSEL1 CKSEL2 CKSEL3 CKSEL4
CKOUT CKDIV8
CLKPCE
CLKPR
CLKPS0 CLKPS1 CLKPS2 CLKPS3
COM0A0 COM0A1
COM0B0 COM0B1
COM1A0 COM1A1
COM1B0 COM1B1
COM2A0 COM2A1
COM2B0 COM2B1
```

```
CPHA
CPOL
CS00 CS01 CS02
CS10 CS11 CS12
CS20 CS21 CS22
DDRB DDB0 DDB1 DDB2 DDB3 DDB4 DDB5 DDB6 DDB7
DDRC DDC0 DDC1 DDC2 DDC3 DDC4 DDC5 DDC6
DDRD DDD0 DDD1 DDD2 DDD3 DDD4 DDD5 DDD6 DDD7
DIDR0 DIDR1
DOR0 DORD
DWEN
E2END
EEPROMEND
EEADRBITS
EE_RDY
EEAR
EEARH EEAR8
EEARL EEAR0 EEAR1 EEAR2 EEAR3 EEAR4 EEAR5 EEAR6 EEAR7
EECR
EEDR EEDR0 EEDR1 EEDR2 EEDR3 EEDR4 EEDR5 EEDR6 EEDR7
EEMPE
EEPE
EEPM0 EEPM1
EERE
EERIE
EESAVE
EICRA
EIFR
EIMSK
ERDYaddr
EXCLK
EXT_INT0 EXT_INT1
EXTRF EXTREF
FE0
FIRSTBOOTSTART FOURTHBOOTSTART
FLASHEND
FOC0A FOC0B
FOC1A FOC1B
FOC2A FOC2B
GPIOR0 GPIOR00 GPIOR01 GPIOR02 GPIOR03 GPIOR04 GPIOR05 GPIOR06 GPIOR07
GPIOR1 GPIOR10 GPIOR11 GPIOR12 GPIOR13 GPIOR14 GPIOR15 GPIOR16 GPIOR17
GPIOR2 GPIOR20 GPIOR21 GPIOR22 GPIOR23 GPIOR24 GPIOR25 GPIOR26 GPIOR27
GTCCR
ICES1
ICF1
ICIE1
ICNC1
ICP1addr
ICR1H ICR1L
INT0 INT1 INT0addr INT1addr
INTF0 INTF1
INT_VECTORS_SIZE
IOEND
ISC00 ISC01
ISC10 ISC11
IVCE
IVSEL
LARGEBOOTSTART
LB1 LB2
MCUCR
MCUSR
MPCM0
MSTR
```

```
MUX0 MUX1 MUX2 MUX3
NRWW_START_ADDR NRWW_STOP_ADDR
OC0Aaddr OC0Baddr OC1Aaddr OC1Baddr OC2Aaddr OC2Baddr
OCF0A OCF0B
OCF1A OCF1B
OCF2A OCF2B
OCIE0A OCIE0B
OCIE1A OCIE1B
OCIE2A OCIE2B
OCR0A OCR0A_0 OCR0A_1 OCR0A_2 OCR0A_3 OCR0A_4 OCR0A_5 OCR0A_6 OCR0A_7
OCR0B OCR0B_0 OCR0B_1 OCR0B_2 OCR0B_3 OCR0B_4 OCR0B_5 OCR0B_6 OCR0B_7
OCR1AH OCR1AL
OCR1BH OCR1BL
OCR2A OCR2A_0 OCR2A_1 OCR2A_2 OCR2A_3 OCR2A_4 OCR2A_5 OCR2A_6 OCR2A_7
OCR2AUB
OCR2B OCR2B_0 OCR2B_1 OCR2B_2 OCR2B_3 OCR2B_4 OCR2B_5 OCR2B_6 OCR2B_7
OCR2BUB
OSCCAL
OVF0addr OVF1addr OVF2addr
power_ctrl_reg
PAGESIZE
PCI0addr PCI1addr PCI2addr
PORTB PORTB0 PORTB1 PORTB2 PORTB3 PORTB4 PORTB5 PORTB6 PORTB7
PINB PINB0 PINB1 PINB2 PINB3 PINB4 PINB5 PINB6 PINB7
PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7
PORTC PORTC0 PORTC1 PORTC2 PORTC3 PORTC4 PORTC5 PORTC6
PINC PINC0 PINC1 PINC2 PINC3 PINC4 PINC5 PINC6
PC0 PC1 PC2 PC3 PC4 PC5 PC6
PORTD PORTD0 PORTD1 PORTD2 PORTD3 PORTD4 PORTD5 PORTD6 PORTD7
PIND PIND0 PIND1 PIND2 PIND3 PIND4 PIND5 PIND6 PIND7
PD0 PD1 PD2 PD3 PD4 PD5 PD6 PD7
PCICR
PCIE0 PCIE1 PCIE2
PCIF0 PCIF1 PCIF2
PCIFR
PCINT0 PCINT1 PCINT2 PCINT3 PCINT4 PCINT5 PCINT6 PCINT7 PCINT8 PCINT9
PCINT10 PCINT11 PCINT12 PCINT13 PCINT14 PCINT15 PCINT16 PCINT17 PCINT18
PCINT19 PCINT20 PCINT21 PCINT22 PCINT23
PCMSK0 PCMSK1 PCMSK2
PGERS PGWRT
PORF
PRADC PRR PRSPI
PRTIM0 PRTIM1 PRTIM2
PRTWI PRUSART0
PSRASY PSR2
PSRSYNC PSR10
PUD
REFS0 REFS1
RSTDISBL
RWWSB RWWSRE
RWW_START_ADDR RWW_STOP_ADDR
RXB80 RXC0 RXCIE0 RXEN0
SECONDBOOTSTART SMALLBOOTSTART
SE SELFPRGEN
SIGNATURE_000 SIGNATURE_001 SIGNATURE_002
SM0 SM1 SM2
SMCR
SPCR
SPDR SPDR0 SPDR1 SPDR2 SPDR3 SPDR4 SPDR5 SPDR6 SPDR7
SPE SPH
SPIaddr SPMRaddr
SPIEN
SPI_STC SPI2X SPIE SPIF SPL SPM_READY SPMCSR SPMIE SPR0 SPR1 SPSR
```

```
SRAM_START SRAM_SIZE
SREG SREG_C SREG_Z SREG_N SREG_V SREG_S SREG_H SREG_T SREG_I
SUT0 SUT1
smcr
TCCR0A TCCR0B
TCCR1A TCCR1B TCCR1C
TCCR2A TCCR2B
TCN2UB
TCNT0 TCNT0_0 TCNT0_1 TCNT0_2 TCNT0_3 TCNT0_4 TCNT0_5 TCNT0_6 TCNT0_7
TCNT1H TCNT1L
TCNT2 TCNT2_0 TCNT2_1 TCNT2_2 TCNT2_3 TCNT2_4 TCNT2_5 TCNT2_6 TCNT2_7
TCR2AUB TCR2BUB
TIFR0 TIFR1 TIFR2
TIM0_COMPA TIM0_COMPB TIM0_OVF
TIM1_CAPT TIM1_COMPA TIM1_COMPB TIM1_OVF
TIM2_COMPA TIM2_COMPB TIM2_OVF
TIMSK0 TIMSK1 TIMSK2
THIRDBOOTSTART
TOIE0 TOIE1 TOIE2 TOIE2A
TOV0 TOV1 TOV2
TSM TWA0 TWA1 TWA2 TWA3 TWA4 TWA5 TWA6
TWAM0 TWAM1 TWAM2 TWAM3 TWAM4 TWAM5 TWAM6
TWAMR0 TWAMR1 TWAMR2 TWAMR3 TWAMR4 TWAMR5 TWAMR6
TWAMR TWAR
TWBR TWBR0 TWBR1 TWBR2 TWBR3 TWBR4 TWBR5 TWBR6 TWBR7
TWCR
TWDR TWD0 TWD1 TWD2 TWD3 TWD4 TWD5 TWD6 TWD7
TWEA TWEN TWGCE
TWI TWIE TWINT TWIaddr
TWPS0 TWPS1 TWS3 TWS4 TWS5 TWS6 TWS7
TWSR TWSTA TWSTO TWWC
TXB80 TXC0 TXCIE0 TXEN0
U2X0
UBRR0H UBRR0L
UBRR0 UBRR1 UBRR2 UBRR3 UBRR4 UBRR5 UBRR6 UBRR7 UBRR8 UBRR9 UBRR10 UBRR11
UCPHA0 UCPOL0 UCSR0A UCSR0B UCSR0C UCSZ00 UCSZ01 UCSZ02
UDORD0 UDR0 UDRE0 UDRIE0
UMSEL00 UMSEL01 UMSEL0 UMSEL1
UDR0_0 UDR0_1 UDR0_2 UDR0_3 UDR0_4 UDR0_5 UDR0_6 UDR0_7
UPE0 UPM00 UPM01
URXCaddr UDREaddr UTXCaddr
USART_DRE USART_RXC USART_TXC USBS0
WCOL
WDCE WDE WDIE WDIF WDP0 WDP1 WDP2 WDP3 WDRF
WDT WDTCSR WDTON WDTaddr
WGM00 WGM01 WGM02 WGM10 WGM11 WGM12 WGM13 WGM20 WGM21 WGM22
XRAMEND

/C5"Arbeits-Register"
X XH XL
Y YH YL
Z ZH ZL
r0 r1 r2 r3 r4 r5 r6 r7 r8 r9 r10 r11 r12 r13 r14 r15 r16
r17 r18 r19 r20 r21 r22 r23 r24 r25 r26 r27 r28 r29 r30 r31
RAMPX RAMPY RAMPZ RAMPD
EIND
STACK

/C6"Funktionen"
BYTE2 BYTE3 BYTE4
EXP2
HIGH HWRD
LOW LOG2 LWRD
```

```
PAGE
RAMEND

/C7"Preprozessor-Anweisungen"
** #

/C8"Direktiven"
** .
```

7.4 Wortsammlung für CodeVisionAVR C-Compiler

Die vollständige Wortsammlung für den C-Compiler in der Datei cvavr.uew

```
/L2"CodeVisionAVR" Line Comment = // Block Comment On = /* Block Comment
Off = */ Escape Char = \ String Chars = "' File Extensions = C CPP CC CXX H
HPP AWK GES i lib

/Delimiters = ~!@%^&*()+-=|\/{}[]:;"'<> ,  ?

/C1"Schluesselwoerter"
auto
break bit bool
_Bool __interrupt
case char const continue
default do double defined
else enum extern
float for
goto
if inline int interrupt
long
register return
short signed sizeof sfrb sfrw static struct switch
typedef
union unsigned
void volatile
while

/C2"Symbole"
(
)
[
]
{
}

/C3"Operatoren"
+
-
=
// /
*
%
&
>
<
^
!
|
~
?
```

```
/C4"AVR-spezifische Namen"
ACBG
ACD
ACI
ACIC
ACIE
ACIS0 ACIS1
ACME
ACO
ACSR
ADATE
ADC_INT
ADC0D ADC1D ADC2D ADC3D ADC4D ADC5D
ADCH ADCL
ADCSRA ADCSRB
ADCW
ADEN
ADIE
ADIF
ADLAR
ADMUX
ADPS0 ADPS1 ADPS2
ADSC
ADTS0 ADTS1 ADTS2
AIN0D AIN1D
ANA_COMP
AS2
ASSR
BLBSET
BODS
BODSE
BORF
CLKPCE
CLKPR
CLKPS0 CLKPS1 CLKPS2 CLKPS3
COM0A0 COM0A1
COM0B0 COM0B1
COM1A0 COM1A1
COM1B0 COM1B1
COM2A0 COM2A1
COM2B0 COM2B1
CPHA
CPOL
CS00 CS01 CS02
CS10 CS11 CS12
CS20 CS21 CS22
DDRB DDB0 DDB1 DDB2 DDB3 DDB4 DDB5 DDB6 DDB7
DDRC DDC0 DDC1 DDC2 DDC3 DDC4 DDC5 DDC6
DDRD DDD0 DDD1 DDD2 DDD3 DDD4 DDD5 DDD6 DDD7
DIDR0 DIDR1
DOR0
DORD
EE_RDY
EEAR
EEARH EEARL
EECR
EEDR
EEMPE
EEPE
EEPM0 EEPM1
EERE
EERIE
EICRA
```

```
EIFR
EIMSK
EXCLK
EXT_INT0 EXT_INT1
EXTRF
FE0
FOC0A FOC0B
FOC1A FOC1B
FOC2A FOC2B
GPIOR0 GPIOR1 GPIOR2
GTCCR
ICES1
ICF1
ICIE1
ICNC1
ICR1H ICR1L
INT0 INT1
INTF0 INTF1
ISC00 ISC01
ISC10 ISC11
IVCE
IVSEL
MCUCR
MCUSR
MPCM0
MSTR
MUX0 MUX1 MUX2 MUX3
OCF0A OCF0B
OCF1A OCF1B
OCF2A OCF2B
OCIE0A OCIE0B
OCIE1A OCIE1B
OCIE2A OCIE2B
OCR0A OCR0B
OCR1AH OCR1AL
OCR1BH OCR1BL
OCR2A OCR2AUB
OCR2B OCR2BUB
OSCCAL
power_ctrl_reg
PORTB PORTB0 PORTB1 PORTB2 PORTB3 PORTB4 PORTB5 PORTB6 PORTB7
PINB PINB0 PINB1 PINB2 PINB3 PINB4 PINB5 PINB6 PINB7
PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7
PORTC PORTC0 PORTC1 PORTC2 PORTC3 PORTC4 PORTC5 PORTC6
PINC PINC0 PINC1 PINC2 PINC3 PINC4 PINC5 PINC6
PC0 PC1 PC2 PC3 PC4 PC5 PC6
PORTD PORTD0 PORTD1 PORTD2 PORTD3 PORTD4 PORTD5 PORTD6 PORTD7
PIND PIND0 PIND1 PIND2 PIND3 PIND4 PIND5 PIND6 PIND7
PD0 PD1 PD2 PD3 PD4 PD5 PD6 PD7
PCICR
PCIE0 PCIE1 PCIE2
PCIF0 PCIF1 PCIF2
PCIFR
PCINT0 PCINT1 PCINT2
PCMSK0 PCMSK1 PCMSK2
PGERS PGWRT
PORF
PRADC PRR PRSPI
PRTIM0 PRTIM1 PRTIM2
PRTWI PRUSART0 PSRASY PSRSYNC PUD
REFS0 REFS1
RWWSB RWWSRE
RXB80 RXC0 RXCIE0 RXEN0
```

```
SE SELFPRGEN
SM0 SM1 SM2
SMCR
SPCR SPDR
SPE SPH
SPI_STC SPI2X SPIE SPIF SPL SPM_READY SPMCSR SPMIE SPR0 SPR1 SPSR
SREG
smcr
TCCR0A TCCR0B
TCCR1A TCCR1B TCCR1C
TCCR2A TCCR2B
TCN2UB
TCNT0
TCNT1H TCNT1L
TCNT2
TCR2AUB TCR2BUB
TIFR0 TIFR1 TIFR2
TIM0_COMPA TIM0_COMPB TIM0_OVF
TIM1_CAPT TIM1_COMPA TIM1_COMPB TIM1_OVF
TIM2_COMPA TIM2_COMPB TIM2_OVF
TIMSK0 TIMSK1 TIMSK2
TOIE0 TOIE1 TOIE2
TOV0 TOV1 TOV2
TSM TWA0 TWA1 TWA2 TWA3 TWA4 TWA5 TWA6
TWAM0 TWAM1 TWAM2 TWAM3 TWAM4 TWAM5 TWAM6
TWAMR TWAR TWBR TWCR TWDR TWEA TWEN TWGCE TWI TWIE TWINT
TWPS0 TWPS1 TWS3 TWS4 TWS5 TWS6 TWS7
TWSR TWSTA TWSTO TWWC
TXB80 TXC0 TXCIE0 TXEN0
U2X0
UBRR0H UBRR0L
UCPHA0 UCPOL0 UCSR0A UCSR0B UCSR0C UCSZ00 UCSZ01 UCSZ02
UDORD0 UDR0 UDRE0 UDRIE0 UMSEL00 UMSEL01
UPE0 UPM00 UPM01 USART_DRE USART_RXC USART_TXC USBS0
WCOL WDCE WDE WDIE WDIF WDP0 WDP1 WDP2 WDP3 WDRF WDT WDTCSR
WGM00 WGM01 WGM02 WGM10 WGM11 WGM12 WGM13 WGM20 WGM21 WGM22
__CODEVISIONAVR__
__STDC__
__LINE__
__FILE__
__TIME__
__UNIX_TIME__
__DATE__
__BUILD__
_MCU_CLOCK_FREQUENCY_
_MODEL_TINY_
_MODEL_SMALL_
_MODEL_MEDIUM_
_MODEL_LARGE_
_OPTIMIZE_SIZE_
_OPTIMIZE_SPEED_
_WARNINGS_ON_
_WARNINGS_OFF_
_PROMOTE_CHAR_TO_INT_ON_
_PROMOTE_CHAR_TO_INT_OFF_
_ENHANCED_CORE_
_HEAP_START_
_HEAP_SIZE_
_UNSIGNED_CHAR_
_8BIT_ENUMS_
__POWERDOWN_SUPPORTED__
__POWERSAVE_SUPPORTED__
__se_bit
```

```
__SLEEP_DEFINED__
__SLEEP_SUPPORTED__
__sm_adc_noise_red
__sm_mask
__sm_powerdown
__sm_powersave
__sm_standby
__STANDBY_SUPPORTED__

/C5"Speicherdefinitionen"
eeprom
flash
__eeprom __flash

/C6"Typ-Synonyme"
BOOL
fpU08
pS8 pS08 pU8 pU08 pS16 pU16 pS32 pU32 pfloat pBOOL
S8 S08 S16 S32
tS08 tU08 tS16 tU16 tS32 tU32
U8 U08 U16 U32
uMEM16 uMEM32

/C7"Preprozessor-Anweisungen"
** #

/C8"C++ Keywords"
catch class const_cast clear
CLEAR
delete dynamic_cast
explicit export
false friend
FALSE
inline
mutable
new namespace
operator on off
ON OFF
private protected public
reinterpret_cast
static_cast set
SET
template this throw true try typeid typename
TRUE
using
virtual
wchar_t
__multiple_inheritance __single_inheritance __virtual_inheritance
```