

# j8051

Een 8051 assembler met syntax highlighting specifiek voor de XC888

## 1. Over j8051

j8051 wordt gemaakt door Dries007, met de intentie om het programma dat op Campus De Nayer werd gebruikt, Reads51 gemaakt door Rigelcorp, te vervangen. In versies 1.1 en 1.2 werd ook functionaliteit toegevoegd om het laden van HEX files naar de XC888 microcontroller over te nemen van FLOAD, die werkt ook op Linux systemen.

Deze assembler heeft geen ondersteuning voor relatief geadresseerde segmenten, maar wel voor absoluut geadresseerde segmenten en include files.

Dit maakt j8051 een all-in-one, cross platform, oplossing voor simpele projecten en onderwijs.

## 2. Licentie en credit

De source code van j8051 is beschikbaar onder de “Modified BSD” licentie op Github<sup>1</sup>, zie bijlage A. De syntax highlighting voor XC888 (voornamelijk de registerdefinities) is gebaseerd op een bestand gemaakt door Roggemans.

## 3. Compatibiliteit

Aangezien j8051 werd geschreven als vervanging voor Reads51 is de syntax gebaseerd op de documentatie van dit programma<sup>2</sup>. De ingebouwde functie om code te uploaden naar de microcontroller is specifiek geschreven voor de XC888 maar is eenvoudig om te bouwen of aangevuld voor andere modellen die gebruik maken van seriële communicatie.

## 4. Revisiegeschiedenis

**V 1.0** - Eerste uitgave

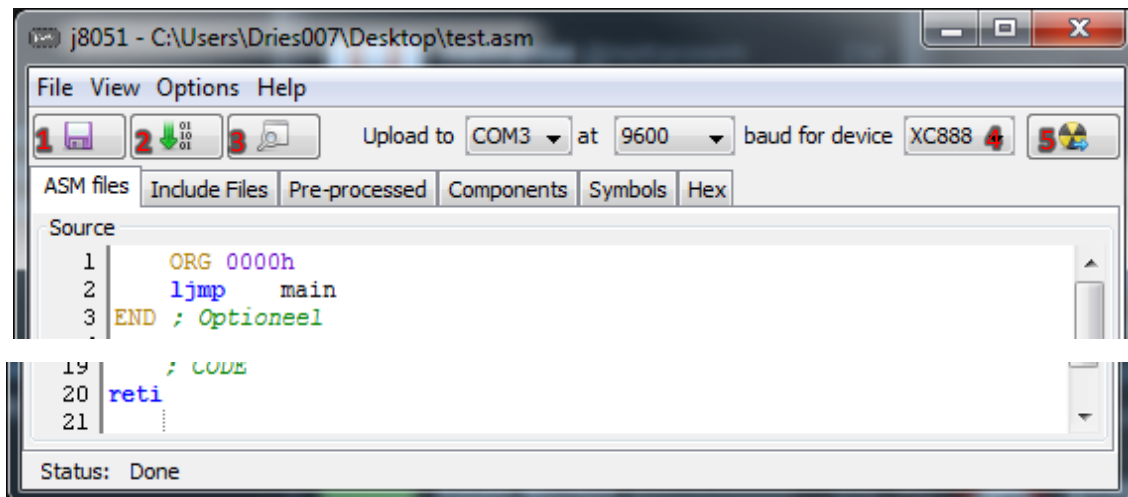
Schrijvers: Dries007

---

<sup>1</sup> <https://github.com/dries007/j8051>

<sup>2</sup> [http://www.rigelcorp.com/\\_\\_doc/8051/reads51.pdf](http://www.rigelcorp.com/__doc/8051/reads51.pdf)

## 5. Overzicht



### 5.1. Menu

NAAM	OPTIES
<b>FILE</b>	New, Open, Save, Compile, Encoding
<b>VIEW</b>	Font, Tab size
<b>OPTIONS</b>	Include folder, Auto save*, Auto compile*
<b>HELP</b>	About

\* = standaard geactiveerd

### 5.2. Werkbalk

	NAAM	SNELTOETS	INFORMATIE
1	Save	Ctrl + S	
2	Compile	F5	
3	Search	Ctrl + F of Ctrl + R	
4	Device type		XC888 is de enige optie op dit moment (v1.2)
5	Upload	F6	

### 5.3. Tabs

NAAM	INFORMATIE
<b>ASM FILES</b>	Weergave van het huidige bestand
<b>INCLUDE FILES</b>	Weergave van alle gebruikte include files
<b>PRE-PROCESSED</b>	Output van de preprocessor
<b>COMPONENTS</b>	Lijst van alle gevonden instructies, lijnummers, adressen en bytecodes
<b>SYMBOLS</b>	Lijst van alle symbolen en lables
<b>HEX</b>	De bytecode output, voorgesteld als tabel

## 6. Syntax

### 6.1. Preprocessor

Alle preprocessor instructies beginnen met een '#' symbool en de syntax is sterk gebaseerd op C.

#### **#include "bestand"**

Voegt het bestand in op deze plaats. De bestandslocatie is, indien van toepassing, relatief t.o.v. de locatie van j8051.

#### **#include <bestand>**

Voegt het bestand in op deze plaats. De bestandslocatie is relatief t.o.v. de "include folder", die via het "Options" menu kan worden ingesteld. Indien er geen "include folder" werd ingesteld, is de locatie relatief t.o.v. de map waarin het bestand staat waaraan gewerkt word.

#### **#define**

Maakt een macro aan, met functionaliteit identiek aan die van C preprocessor macro's. Er zijn verschillende vormen mogelijk:

Simpele tekst vervanging, bijvoorbeeld:

```
#define MAX 10
```

Dit stelt het woord "MAX" gelijk aan "10". Ook macro's met parameter worden ondersteund, bijvoorbeeld:

```
#undefine ADD(a,b) (a + b)
```

#### **#undefine**

Maakt een macro definitie ongedaan.

#### **#ifdef, #ifndef, #else en #endif**

De werking, die gelijk is aan die van de C preprocessor, kan best worden geïllustreerd met het volgende voorbeeld:

```
#define DEBUG

#ifdef DEBUG
mov a, TMOD
#else
clr a
#endif
```

De code

```
mov a, TMOD
```

zal wel worden omgezet naar bytecode maar de lijn

```
clr a
```

niet.

## 6.2. Constanten

Getallen zonder pre of suffix worden aanzien als decimale getallen. De tabel hieronder maakt de andere mogelijkheden duidelijk. Zowel de hexadecimale constante als de pre en suffixen zijn niet hoofdlettergevoelig.

	Prefix	Suffix	Voorbeeld
Hexadecimaal	0x	H	aaH = 0xAA = 170
Octaal		O	55o = 45
Binaire		B	111000b = 56

Voor ASCII constanten worden enkele aanhalingstekens gebruikt, voorbeeld:

'a'

Voor teksten worden dubbele aanhalingstekens gebruikt, voorbeeld:

"Voorbeeldtekstje"

Er wordt geen null byte toegevoegd aan het einde van de string!

## 6.3. Wiskundige uitdrukkingen

Ook bij wiskundige uitdrukkingen is de syntax gelijk aan die van C. Haakjes worden gebruikt om de volgorde van bewerkingen te beïnvloeden. In alle wiskundige uitdrukkingen kunnen labels gebruikt worden als waarde.

Symbol	Voorbeeld	Naam
+	mov a, #(1+2)	Optelling
-	mov a, #(1-2)	Aftrekking
*	mov a, #(3*2)	Vermenigvuldiging
/	mov a, #(8/2)	Deling
%	mov a, #(7%3)	Restdeling
<<	mov a, #(1<<2)	Shift naar links
>>	mov a, #(0x100>>2)	Shift naar rechts
&	mov a, #(1&2)	Binaire AND
	mov a, #(1 2)	Binaire OR
^	mov a, #(1^2)	Binaire XOR
~	mov a, #(~2)	Binaire NOT
&&	mov a, #(1&&2)	Logische AND
	mov a, #(1  2)	Logische OR
!	mov a, #(!1)	Logische NOT
==	mov a, #(1==2)	Is gelijk aan
!=	mov a, #(1!=2)	Is niet gelijk aan
<	mov a, #(1<2)	Is kleiner dan
>	mov a, #(1>2)	Is groter dan
<=	mov a, #(1<=2)	Is kleiner dan of gelijk aan
>=	mov a, #(1>=2)	Is groter dan of gelijk aan
.	setb TCON.1	Bit van register

## 6.4. Functies

Er zijn 2 functies beschikbaar: HIGH en LOW

### HIGH(N)

Deze functie is functioneel equivalent aan  $(N \gg 8)$ , m.a.w. de hoogste byte van een 2 byte getal.

### LOW(N)

Deze functie is functioneel equivalent aan  $(N \& 0xFF)$ , m.a.w. de laagste byte van een 2 byte getal.

## 6.5. Assembler instructies

Er kunnen wiskundige berekeningen gebruikt worden. De volgende instructies zijn beschikbaar om de assembler aan te sturen:

### 6.5.1. Constanten

#### EQU, DATA, BIT

Stelt een symbool gelijk aan een getal. Positie in de code maakt niet uit. Er wordt geen controle uitgevoerd op het gebruik van dit symbool.

Voorbeeld:

```
Mov    a, TELLER
TELLER EQU 28h
```

### 6.5.2. Data en tabellen

#### DB

Definieer bytes handmatig.

Voorbeeld:

```
DB      0, 1, 2, 3          ; Resultaat: 0x00 0x01 0x02 0x03
DB      "Hello", 0x00       ; Resultaat: 0x48 0x65 0x6c 0x6c 0x6f 0x00
```

#### DW

Definieer words (2 bytes) handmatig. Indien de waarde kleiner is dan 1 byte, word een extra null byte toegevoegd.

Voorbeeld:

```
DW      1234h              ; Resultaat: 0x1234h
DW      "Test", 00h        ; Resultaat: 0x0054 0x0065 0x0073 0x0074 0x0000
```

#### DS

Definieer een block met aangeduide grootte. Initialiseren naar een bepaalde waarde is optioneel.

Voorbeeld:

```
DS      5                  ; Resultaat: 0x00 0x00 0x00 0x00 0x00
DS      5 << 0xFF          ; Resultaat: 0xFF 0xFF 0xFF 0xFF 0xFF
```

### 6.5.3. Segmenten

#### ORG

Start een segment op een bepaalde locatie.

Voorbeeld:

```
ORG 0000h
```

#### END

Beëindig een segment. Het gebruik van deze instructie is niet verplicht.

Voorbeeld van ORG en END

```
ORG 0000h
ljmp main
END ; Optioneel

ORG 0003h
ljmp interupt1
END ; Optioneel

ORG 0006h
ljmp interupt2

main:
    ; CODE
    ljmp main
interupt1:
    ; CODE
    reti
interupt2:
    ; CODE
    reti
```

## 7. Bijlage A

De "Modified BSD" licentie

Copyright (c) 2014, Dries007

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.