

«

»

**1-31 03 07**

**1-31 03 07-02**

( )

(

)

2024 .

:  
 . . , -  
 .  
 :  
 -  
 ( 8 23 2024 .);  
 07-02 ( ( ) ( 1-31 03 07  
 1-31 03 -  
 ))). -  
 , , -  
 8- Microchip AVR® -  
 , -  
 Arduino. ATMEGA -  
 GNU. -  
 AVR C/C++. -

( )

korzhukov@bsu.by 398-02-22  
2024

// 4440 843 ( Portable 20 arduino-1.8.5. 9)  
4420 834.  
AVR-8 -  
[1].

**1**  
Arduino.h – SDK Arduino.  
: setup() loop(). -

**1.1** **.h**  
\\arduino-1.8.5\\hardware\\arduino\\avr\\cores\\arduino  
7 483 Arduino.h  
11 214 binary.h  
1 529 Client.h  
Wiring 5 262 HardwareSerial.h  
4 469 HardwareSerial\_private.h  
2 861 IPAddress.h  
new delete 979 new.h  
2 063 PluggableUSB.h  
2 963 Print.h  
1 335 Printable.h  
963 Server.h  
6 060 Stream.h  
4 363 Udp.h  
6 261 USBAPI.h  
8 435 USBCore.h  
1 519 USBDesc.h  
4 576 WCharacter.h  
2 255 wiring\_private.h  
Wiring Arduino 9 910 WString.h  
19 Ar-  
duino.

\\hardware\\arduino\\avr\\variants 6 . -

Arduino .

\\hardware\\tools\\avr\\avr\\include 17 . -

\\hardware\\tools\\avr\\avr\\include\\avr 288 . -

) AVR-8 ( -  
Atmel.

## 1.2

### **.c .cpp**

\arduino-1.8.5\hardware\arduino\avr\cores\arduino

		1 142 hooks.c
		9 409 WInterrupts.c
Wiring API	ATmega	12 024 wiring.c
		7 850 wiring_analog.c
		4 978 wiring_digital.c
		3 435 wiring_pulse.c
		1 550 wiring_shift.c

7

Arduino

\arduino-1.8.5\hardware\arduino\avr\cores\arduino

		1 222 abi.cpp
		8 078 CDC.cpp
		7 743 HardwareSerial.cpp
		2 605 HardwareSerial0.cpp
		2 315 HardwareSerial1.cpp
		1 975 HardwareSerial2.cpp
		1 975 HardwareSerial3.cpp
		2 851 IPAddress.cpp
		1 372 main.cpp
		1 027 new.cpp
		2 725 PluggableUSB.cpp
		5 442 Print.cpp
		8 804 Stream.cpp
		15 022 Tone.cpp
		20 086 USBCore.cpp
		1 641 WMath.cpp
Wiring	Arduino	16 989 WString.cpp

17

Arduino

++.

### **main.cpp**

```
#include <Arduino.h>

int atexit(void (* /*func*/ )()) { return 0; }

void initVariant() __attribute__((weak));
void initVariant() { }

void setupUSB() __attribute__((weak));
void setupUSB() { }

int main(void) {
    init();
    initVariant();
    #if defined(USBCON)
        USBDevice.attach();
    #endif
    setup();
    for (;;) {
        loop();
        if (serialEventRun) serialEventRun();
    }
    return 0;
}
```

## 2

### **.EXE**

\arduino-1.8.5:

3 291 136	arduino-builder.exe
404 480	arduino.exe

\arduino-1.8.5\hardware\tools\avr\avr\bin:

741 888	ar.exe
741 888	ranlib.exe
956 416	as.exe
1 258 496	ld.bfd.exe
1 258 496	ld.exe
727 040	nm.exe
892 416	objcopy.exe
1 026 048	objdump.exe
514 048	readelf.exe
892 416	strip.exe

( )

ELF

10 .

\arduino-1.8.5\hardware\tools\avr\bin:

6 619 136	arduinoOTA.exe
716 288	avr-addr2line.exe
741 888	avr-ar.exe
956 416	avr-as.exe
C++ 788 480	avr-c++.exe
714 752	avr-c++filt.exe
786 944	avr-cpp.exe
69 632	avr-elfedit.exe
C++ 788 480	avr-g++.exe
785 408	avr-gcc-4.9.2.exe
65 024	avr-gcc-ar.exe
65 024	avr-gcc-nm.exe
65 024	avr-gcc-ranlib.exe
785 408	avr-gcc.exe
488 448	avr-gcov.exe
3 888 128	avr-gdb.exe
784 896	avr-gprof.exe
1 258 496	avr-ld.bfd.exe
1 258 496	avr-ld.exe
727 040	avr-nm.exe
892 416	avr-objcopy.exe
1 026 048	avr-objdump.exe
741 888	avr-ranlib.exe
514 048	avr-readelf.exe
837 120	avr-run.exe
726 016	avr-size.exe
716 800	avr-strings.exe
892 416	avr-strip.exe
524 800	avrdude.exe
17 408	loaddrv.exe

30 .

/ ++.

**avr-as**

The Assembler.

**avr-ld**

The Linker.

( )

**avr-ar**

( ).

**avr-ranlib**

( ).

**avr-objcopy**

**avr-objdump**

**avr-size**

**avr-nm**

**avr-strings**

**avr-strip**

**avr-readelf**

ELF.

**avr-addr2line**

( ).

**avr-c++filt**

C ++.

**avr-libc**

GCC Binutils

:

C.

\arduino-1.8.5\hardware\tools\avr\libexec\gcc\avr\4.9.2:

++

9 208 320 cc1.exe  
10 207 232 cc1plus.exe  
504 320 collect2.exe  
646 144 lto-wrapper.exe  
8 565 760 lto1.exe

5

**cc1**

**cc1plus**

C++.

**collect2**

GNU,

collect2

(global initialization code),

**cc1.exe --help**

**cc1plus.exe --help**

### 3 GNU

c++

gcc,

C++

C++

g++ .

g++

gcc,

C++

C++.

++ .

gcc

(linkers)

**avr-gcc.exe --h**

AVR

## 4

### 4.1

#### Intel HEX HEX

16- ( 16- (4 ) ).

.hex.

I8HEX,

I16HEX I32HEX.  
00/01 (16-  
) , I32HEX

), I16HEX I8HEX  
02/03 (20-  
04/05 (32-  
).

(:).

( ).  
1,

MCS-51:

:02000004FF00FB  
:0E0000007F16EF3395E0FE7DDCED3395E0FCDE  
:00000001FF

1. (:).

2. (2), 16 32 . -

2 3. (4) . -

4. (2) . -

5. ( , 0 255 ) . -

6. 0 . -

7. / (CR/LF - 0Dh 0Ah). -

«:LLAAAATTDD...CC», :

LL ,  
AAAA ,  
TT ,  
DD ( ),  
CC .

- 0 -
- 1 -

«:00000001FF».

- 2 - ( - ) . -
- 3 - . -

80x86 ( CS:IP - ) . -

- 4 - . " " -
- 5 - 32- " " -



## 5 GNU

### 5.1 Objdump

**objdump** (GNU Binutils) –

, , . -  
-  
, , (.hex). -  
(help) -  
:

objdump.exe --help

**avr-objdump.exe objdump.exe**

**objdump**

, , !  
: objdump <option(s)> <file(s)>  
( ). -  
-

**-a, --archive-headers**  
**-f, --file-headers**

**-p, --private-headers**

**-P, --private=OPT,OPT...**

**-h, --[section-]headers**  
( )

**-x, --all-headers**

**-d, --disassemble** ( )

**-D, --disassemble-all** ( )  
( .hex)

**-S, --source** -

**-s, --full-contents** ( )  
( ),

**-g, --debugging**

**-e, --debugging-tags**,

ctags

**-G, --stabs** ( )

STABS

**-W[ILiaprnmfFsoRt]**

--dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,  
=frames-interp,=str,=loc,=Ranges,=pubtypes,  
=gdb\_index,=trace\_info,=trace\_abbrev,=trace\_aranges,  
=addr,=cu\_index]

DWARF

**-t, --syms** ( ) ( -

**-T, --dynamic-syms** -

**-r, --reloc** (

**-R, --dynamic-reloc**  
@<file> <file>

**-v, --version**

**-i, --info** -

**-H, --help**

, objdump -v

GNU objdump (GNU Binutils) 2.26.20160125

Copyright (C) 2015 Free Software Foundation, Inc.

This program is free software; you may redistribute it under the terms of the GNU General Public License version 3 or (at your option) any later version.

This program has absolutely no warranty.

, objdump -i

BFD header file version (GNU Binutils) 2.26.20160125

elf32-avr

(header little endian, data little endian)

avr

elf32-little

(header little endian, data little endian)

plugin

avr

elf32-big

(header big endian, data big endian)

plugin

avr

plugin

(header little endian, data little endian)

srec

(header endianness unknown, data endianness unknown)

plugin

avr

symbolsrec

(header endianness unknown, data endianness unknown)

plugin

avr

verilog

(header endianness unknown, data endianness unknown)

plugin

avr

tekhex

(header endianness unknown, data endianness unknown)

plugin

avr

binary

(header endianness unknown, data endianness unknown)

plugin

avr

ihex

(header endianness unknown, data endianness unknown)

plugin

avr

```
                elf32-avr elf32-little elf32-big plugin srec symbolsrec verilog
plugin  ----- elf32-little elf32-big ----- srec symbolsrec verilog
        avr elf32-avr elf32-little elf32-big ----- srec symbolsrec verilog
```

```
                tekhex binary ihex
plugin tekhex binary ihex
        avr tekhex binary ihex
```

## **objdump -help**      **objdump -H**

:objdump.exe <option(s)> <file(s)>

Display information from object <file(s)>.

At least one of the following switches must be given:

- a, --archive-headers    Display archive header information
- f, --file-headers      Display the contents of the overall file header
- p, --private-headers   Display object format specific file header contents
- P, --private=OPT,OPT... Display object format specific contents
- h, --[section-]headers Display the contents of the section headers
- x, --all-headers       Display the contents of all headers
- d, --disassemble       Display assembler contents of executable sections
- D, --disassemble-all   Display assembler contents of all sections
- S, --source            Intermix source code with disassembly
- s, --full-contents     Display the full contents of all sections requested
- g, --debugging         Display debug information in object file
- e, --debugging-tags    Display debug information using ctags style
- G, --stabs             Display (in raw form) any STABS info in the file
- W[ILiaprnmfFsoRt] or
- dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,  
=frames-interp,=str,=loc,=Ranges,=pubtypes,  
=gdb\_index,=trace\_info,=trace\_abbrev,=trace\_aranges,

=addr,=cu\_index]  
 Display DWARF info in the file  
 -t, --syms Display the contents of the symbol table(s)  
 -T, --dynamic-syms Display the contents of the dynamic symbol table  
 -r, --reloc Display the relocation entries in the file  
 -R, --dynamic-reloc Display the dynamic relocation entries in the file  
 @<file> Read options from <file>  
 -v, --version Display this program's version number  
 -i, --info List object formats and architectures supported  
 -H, --help Display this information

:  
**-b, --target=BFDNAME**  
 BFDNAME  
**-m, --architecture=MACHINE**  
 MACHINE  
**-j, --section=NAME**  
 NAME  
**-M, --disassembler-options=OPT** OPT

**-EB --endian=big**  
 big endian  
**-EL --endian=little**  
 little endian  
**--file-start-context** ( -S)

**-I, --include=DIR** DIR,  
**-l, --line-numbers** Include line numbers and filenames in output  
**-F, --file-offsets** Include file offsets when displaying information  
**-C, --demangle[=STYLE]** Decode mangled/processed symbol names  
 The STYLE, if specified, can be 'auto', 'gnu',  
 'lucid', 'arm', 'hp', 'edg', 'gnu-v3', 'java'  
 or 'gnat'

-w, --wide Format output for more than 80 columns  
 -z, --disassemble-zeroes Do not skip blocks of zeroes when disassembling  
**--start-address=ADDR** Only process data whose address is >= ADDR  
**--stop-address=ADDR** Only process data whose address is <= ADDR  
**--prefix-addresses** Print complete address alongside disassembly  
**--[no-]show-raw-insn** Display hex alongside symbolic disassembly  
**--insn-width=WIDTH** Display WIDTH bytes on a single line for -d  
**--adjust-vma=OFFSET** Add OFFSET to all displayed section addresses  
**--special-syms** Include special symbols in symbol dumps  
**--prefix=PREFIX** Add PREFIX to absolute paths for -S  
**--prefix-strip=LEVEL** Strip initial directory names for -S  
**--dwarf-depth=N** Do not display DIEs at depth N or greater  
 N

--dwarf-start=N      Display DIEs starting with N, at the same depth or deeper  
                                N,  
--dwarf-check      Make additional dwarf internal consistency checks.

objdump: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec veri-  
log tekhex binary ihex

objdump: supported architectures: avr avr:1 avr:2 avr:25 avr:3 avr:31 avr:35 avr:4 avr:5  
avr:51 avr:6 avr:100 avr:101 avr:102 avr:103 avr:104 avr:105 avr:106 avr:107 plugin

Options supported for **-P/--private** switch:      ,      -P/--  
  :

For AVR ELF files:

mem-usage    Display memory usage

avr-prop     Display contents of .avr.prop section

Report bugs to <<http://www.sourceware.org/bugzilla/>>.

## 5.2              readelf.exe

.\arduino-1.8.5\hardware\tools\avr\avr\bin\readelf.exe

readelf.exe 514 048 05.12.2016 17:53

>**readelf --h**

readelf: option '--h' is ambiguous

Usage: readelf <option(s)> elf-file(s)

>**readelf --help**

        : readelf <option(s)> elf-file(s)

Display information about the contents of ELF format files  
                                ELF

Options are:                  :

-**a --all**                  Equivalent to: -h -l -S -s -r -d -V -A -I                  : -h -l -S -s -r -d  
-V -A -I

-**h --file-header**      Display the ELF file header                  ELF

-**l --program-headers**    Display the program headers                  -

-**segments**              An alias for --program-headers                  --

-**S --section-headers**    Display the sections' header

--sections              An alias for --section-headers

**-g --section-groups** Display the section groups  
**-t --section-details** Display the section details  
**-e --headers** Equivalent to: **-h -l -S**  
**-s --syms** Display the symbol table  
**--symbols** An alias for **--syms**  
**--dyn-syms** Display the dynamic symbol table  
**-n --notes** Display the core notes (if present)  
**-r --relocs** Display the relocations (if present)  
**-u --unwind** Display the unwind info (if present)  
**-d --dynamic** Display the dynamic section (if present)  
**-V --version-info** Display the version sections (if present)  
**-A --arch-specific** Display architecture specific information (if any)  
**-c --archive-index** Display the symbol/file index in an archive  
**-D --use-dynamic** Use the dynamic section info when displaying symbols  
**-x --hex-dump=<number|name>**  
Dump the contents of section **<number|name>** as bytes  
**-p --string-dump=<number|name>**  
Dump the contents of section **<number|name>** as strings  
**-R --relocated-dump=<number|name>**  
Dump the contents of section **<number|name>** as relocated  
bytes  
**-z --decompress** Decompress section before dumping it  
**-w[llLiaprmmFFsoRt]** or  
**--debug-**  
**dump[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,**  
**=frames-interp,=str,=loc,=Ranges,=pubtypes,**  
**=gdb\_index,=trace\_info,=trace\_abbrev,=trace\_aranges,**  
**=addr,=cu\_index]**  
Display the contents of DWARF2 debug sections  
**--dwarf-depth=N** Do not display DIEs at depth N or greater  
N  
**--dwarf-start=N** Display DIEs starting with N, at the same depth or deeper  
, N,  
**-I --histogram** Display histogram of bucket list lengths  
-  
**-W --wide** Allow output width to exceed 80 characters  
80  
**@<file>** Read options from **<file>** < >  
**-H --help** Display this information  
**-v --version** Display the version number of readelf  
readelf  
Report bugs to **<<http://www.sourceware.org/bugzilla/>>**

### >readelf -v

GNU readelf (GNU Binutils) 2.26.20160125

Copyright (C) 2015 Free Software Foundation, Inc.

This program is free software; you may redistribute it under the terms of the GNU General Public License version 3 or (at your option) any later version.

This program has absolutely no warranty.

### 5.3 strip.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\strip.exe

strip.exe 892 416 05.12.2016 17:53

#### >strip --help

: strip <option(s)> in-file(s)

Removes symbols and sections from files

The options are:

**-I --input-target=<bfdname>** Assume input file is in format <bfdname> -  
<bfdname>

**-O --output-target=<bfdname>** Create an output file in format <bfdname>  
<bfdname>

**-F --target=<bfdname>** Set both input and output format to <bfdname>  
<bfdname>

-p --preserve-dates Copy modified/access timestamps to the output

-D --enable-deterministic-archives Produce deterministic output when stripping ar-  
chives

-U --disable-deterministic-archives Disable -D behavior (default)

-R --remove-section=<name> Also remove section <name> from the output

-s --strip-all Remove all symbol and relocation information

-g -S -d --strip-debug Remove all debugging symbols & sections

--strip-dwo Remove all DWO sections

--strip-unneeded Remove all symbols not needed by relocations

--only-keep-debug Strip everything but the debug information

-N --strip-symbol=<name> Do not copy symbol <name>

-K --keep-symbol=<name> Do not strip symbol <name>

--keep-file-symbols Do not strip file symbol(s)

-w --wildcard Permit wildcard in symbol comparison

-x --discard-all Remove all non-global symbols

-X --discard-locals Remove any compiler-generated symbols

-v --verbose List all object files modified

-V --version Display this program's version number

-h --help Display this output

--info List object formats & architectures supported

-o <file> Place stripped output into <file>

strip: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec  
verilog tekhex binary ihex  
Report bugs to <<http://www.sourceware.org/bugzilla/>>

#### >strip -V

GNU strip (GNU Binutils) 2.26.20160125

Copyright (C) 2015 Free Software Foundation, Inc.

This program is free software; you may redistribute it under the terms of  
the GNU General Public License version 3 or (at your option) any later version.

This program has absolutely no warranty.

## 5.4 objcopy.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\objcopy.exe

objcopy.exe 892 416 05.12.2016 17:53

### >objcopy --help

: objcopy [option(s)] in-file [out-file]

Copies a binary file, possibly transforming it in the process

The options are:

-I --input-target <bfdname>	Assume input file is in format <bfdname>
-O --output-target <bfdname>	Create an output file in format <bfdname>
-B --binary-architecture <arch>	Set output arch, when input is arch-less
-F --target <bfdname>	Set both input and output format to <bfdname>
--debugging	Convert debugging information, if possible
-p --preserve-dates	Copy modified/access timestamps to the output
-D --enable-deterministic-archives	Produce deterministic output when stripping archives
-U --disable-deterministic-archives	Disable -D behavior (default)
-j --only-section <name>	Only copy section <name> into the output
--add-gnu-debuglink=<file>	Add section .gnu_debuglink linking to <file>
-R --remove-section <name>	Remove section <name> from the output
-S --strip-all	Remove all symbol and relocation information
-g --strip-debug	Remove all debugging symbols & sections
--strip-dwo	Remove all DWO sections
--strip-unneeded	Remove all symbols not needed by relocations
-N --strip-symbol <name>	Do not copy symbol <name>
--strip-unneeded-symbol <name>	Do not copy symbol <name> unless needed by relocations
--only-keep-debug	Strip everything but the debug information
--extract-dwo	Copy only DWO sections
--extract-symbol	Remove section contents but keep symbols
-K --keep-symbol <name>	Do not strip symbol <name>
--keep-file-symbols	Do not strip file symbol(s)
--localize-hidden	Turn all ELF hidden symbols into locals
-L --localize-symbol <name>	Force symbol <name> to be marked as a local
--globalize-symbol <name>	Force symbol <name> to be marked as a global
-G --keep-global-symbol <name>	Localize all symbols except <name>
-W --weaken-symbol <name>	Force symbol <name> to be marked as a weak
--weaken	Force all global symbols to be marked as weak
-w --wildcard	Permit wildcard in symbol comparison
-x --discard-all	Remove all non-global symbols
-X --discard-locals	Remove any compiler-generated symbols
-i --interleave[=<number>]	Only copy N out of every <number> bytes
--interleave-width <number>	Set N for --interleave
-b --byte <num>	Select byte <num> in every interleaved block
--gap-fill <val>	Fill gaps between sections with <val>
--pad-to <addr>	Pad the last section up to address <addr>
--set-start <addr>	Set the start address to <addr>
{--change-start --adjust-start} <incr>	Add <incr> to the start address
{--change-addresses --adjust-vma} <incr>	Add <incr> to LMA, VMA and start addresses



```

{--change-section-address|--adjust-section-vma} <name>{=|+|-}<val>
    Change LMA and VMA of section <name> by <val>

--change-section-lma <name>{=|+|-}<val>
    Change the LMA of section <name> by <val>
--change-section-vma <name>{=|+|-}<val>
    Change the VMA of section <name> by <val>
{--[no-]change-warnings|--[no-]adjust-warnings}
    Warn if a named section does not exist
--set-section-flags <name>=<flags>
    Set section <name>'s properties to <flags>
--add-section <name>=<file>
    Add section <name> found in <file> to output
--update-section <name>=<file>
    Update contents of section <name> with contents
found in <file>
--dump-section <name>=<file>
    Dump the contents of section <name> into <file>
--rename-section <old>=<new>[,<flags>]
    Rename section <old> to <new>
--long-section-names {enable|disable|keep}
    Handle long section names in Coff objects.
--change-leading-char
    Force output format's leading character style

--remove-leading-char
    Remove leading character from global symbols
--reverse-bytes=<num>
    Reverse <num> bytes at a time, in output sections
with content
--redefine-sym <old>=<new>
    Redefine symbol name <old> to <new>
--redefine-syms <file>
    --redefine-sym for all symbol pairs
    listed in <file>
--srec-len <number>
    Restrict the length of generated Srecords
--srec-forceS3
    Restrict the type of generated Srecords to S3

--strip-symbols <file>
    -N for all symbols listed in <file>
--strip-unneeded-symbols <file>
    --strip-unneeded-symbol for all symbols listed
in <file>
--keep-symbols <file>
    -K for all symbols listed in <file>
--localize-symbols <file>
    -L for all symbols listed in <file>
--globalize-symbols <file>
    --globalize-symbol for all in <file>
--keep-global-symbols <file>
    -G for all symbols listed in <file>
--weaken-symbols <file>
    -W for all symbols listed in <file>
--add-symbol <name>=[<section>:]<value>[,<flags>]
    Add a symbol
--alt-machine-code <index>
    Use the target's <index>'th alternative machine
--writable-text
    Mark the output text as writable
--readonly-text
    Make the output text write protected
--pure
    Mark the output file as demand paged
--impure
    Mark the output file as impure
--prefix-symbols <prefix>
    Add <prefix> to start of every symbol name
--prefix-sections <prefix>
    Add <prefix> to start of every section name
--prefix-alloc-sections <prefix>
    Add <prefix> to start of every allocatable section
name
--file-alignment <num>
    Set PE file alignment to <num>
--heap <reserve>[,<commit>]
    Set PE reserve/commit heap to <reserve>/<commit>
--image-base <address>
    Set PE image base to <address>
--section-alignment <num>
    Set PE section alignment to <num>
--stack <reserve>[,<commit>]
    Set PE reserve/commit stack to <reserve>/<commit>
--subsystem <name>[:<version>]
    Set PE subsystem to <name> [& <version>]
--compress-debug-sections[={none|zlib|zlib-gnu|zlib-gabi}]
    Compress DWARF debug sections using zlib
--decompress-debug-sections
    Decompress DWARF debug sections using zlib
-v --verbose
    List all object files modified
@<file>
    Read options from <file>
-V --version
    Display this program's version number
-h --help
    Display this output
--info
    List object formats & architectures supported

```

objcopy: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec  
verilog tekhex binary ihex  
Report bugs to <<http://www.sourceware.org/bugzilla/>>

## >objcopy -V

GNU objcopy (GNU Binutils) 2.26.20160125

Copyright (C) 2015 Free Software Foundation, Inc.

This program is free software; you may redistribute it under the terms of the GNU General Public License version 3 or (at your option) any later version.

This program has absolutely no warranty.

## 5.5 nm.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\nm.exe

nm.exe 727 040 05.12.2016 17:53

### >nm --help

:nm [option(s)] [file(s)]

List symbols in [file(s)] (a.out by default). [ ( )]  
(a.out ).

The options are:

-a, --debug-syms	Display debugger-only symbols
-A, --print-file-name	Print name of the input file before every symbol
-B	Same as --format=bsd
-C, --demangle[=STYLE]	Decode low-level symbol names into user-level names The STYLE, if specified, can be `auto' (the default), `gnu', `lucid', `arm', `hp', `edg', `gnu-v3', `java' or `gnat'
--no-demangle	Do not demangle low-level symbol names
-D, --dynamic	Display dynamic symbols instead of normal symbols
--defined-only	Display only defined symbols
-e	(ignored)
-f, --format=FORMAT	Use the output format FORMAT. FORMAT can be `bsd', `sysv' or `posix'. The default is `bsd'
-g, --extern-only	Display only external symbols
-l, --line-numbers	Use debugging information to find a filename and line number for each symbol
-n, --numeric-sort	Sort symbols numerically by address
-o	Same as -A
-p, --no-sort	Do not sort the symbols
-P, --portability	Same as --format=posix
-r, --reverse-sort	Reverse the sense of the sort
--plugin NAME	Load the specified plugin
-S, --print-size	Print size of defined symbols
-s, --print-armap	Include index for symbols from archive members
--size-sort	Sort symbols by size
--special-syms	Include special symbols in the output
--synthetic	Display synthetic symbols as well
-t, --radix=RADIX	Use RADIX for printing symbol values
--target=BFDNAME	Specify the target object format as BFDNAME
-u, --undefined-only	Display only undefined symbols
-X 32_64	(ignored)
@FILE	Read options from FILE
-h, --help	Display this information
-V, --version	Display this program's version number

nm: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec ver-ilog tekhex binary ihex  
Report bugs to <<http://www.sourceware.org/bugzilla/>>.

## 5.6 ar.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\ar.exe

ar.exe 741 888 05.12.2016 17:53

### >ar --help

Usage: ar [emulation options] [-]{dmpqrstx}[abcDfilMNOpsSTuvV] [--plugin <name>] [member-name] [count] archive-file file...

ar -M [<mri-script>]

commands:

d - delete file(s) from the archive  
m[ab] - move file(s) in the archive  
p - print file(s) found in the archive  
q[f] - quick append file(s) to the archive  
r[ab][f][u] - replace existing or insert new file(s) into the archive  
s - act as ranlib  
t - display contents of archive  
x[o] - extract file(s) from the archive

command specific modifiers:

[a] - put file(s) after [member-name]  
[b] - put file(s) before [member-name] (same as [i])  
[D] - use zero for timestamps and uids/gids  
[U] - use actual timestamps and uids/gids (default)  
[N] - use instance [count] of name  
[f] - truncate inserted file names  
[P] - use full path names when matching  
[o] - preserve original dates  
[u] - only replace files that are newer than current archive contents

generic modifiers:

[c] - do not warn if the library had to be created  
[s] - create an archive index (cf. ranlib)  
[S] - do not build a symbol table  
[T] - make a thin archive  
[v] - be verbose  
[V] - display the version number  
@<file> - read options from <file>  
--target=BFDNAME - specify the target object format as BFDNAME

optional:

--plugin <p> - load the specified plugin

emulation options:

No emulation specific options

ar: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec ver-ilog tekhex binary ihex  
Report bugs to <<http://www.sourceware.org/bugzilla/>>

## 5.7 ranlib.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\ranlib.exe

ranlib.exe 741 888 05.12.2016 17:53

## >ranlib --help

Usage: ranlib [options] archive

Generate an index to speed access to archives

The options are:

@<file>	Read options from <file>
--plugin <name>	Load the specified plugin
-D	Use zero for symbol map timestamp
-U	Use actual symbol map timestamp (default)
-t	Update the archive's symbol map timestamp
-h --help	Print this help message
-v --version	Print version information

ranlib: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec  
verilog tekhex binary ihex

Report bugs to <<http://www.sourceware.org/bugzilla/>>

## 5.8 as.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\as.exe

as.exe 956 416 05.12.2016 17:54

## >as --help

Usage: as [option...] [asmfile...]

Options:

-a[sub-option...]	turn on listings
	Sub-options [default hls]:
	c omit false conditionals
	d omit debugging directives
	g include general info
	h include high-level source
	l include assembly
	m include macro expansions
	n omit forms processing
	s include symbols
	=FILE list to FILE (must be last sub-option)
--alternate	initially turn on alternate macro syntax
--compress-debug-sections[={none zlib zlib-gnu zlib-gabi}]	compress DWARF debug sections using zlib
--nocompress-debug-sections	don't compress DWARF debug sections [default]
-D	produce assembler debugging messages
--debug-prefix-map OLD=NEW	map OLD to NEW in debug information
--defsym SYM=VAL	define symbol SYM to given value
--execstack	require executable stack for this object
--noexecstack	don't require executable stack for this object
--size-check=[error warning]	ELF .size directive check (default --size-check=error)
--sectname-subst	enable section name substitution sequences
-f	skip whitespace and comment preprocessing
-g --gen-debug	generate debugging information
--gstabs	generate STABS debugging information
--gstabs+	generate STABS debug info with GNU extensions
--gdwarf-2	generate DWARF2 debugging information
--gdwarf-sections	generate per-function section names for DWARF line information
--hash-size=<value>	set the hash table size close to <value>
--help	show this message and exit

```

--target-help          show target specific options
-I DIR                add DIR to search list for .include directives
-J                    don't warn about signed overflow
-K                    warn when differences altered for long displacements
-L,--keep-locals      keep local symbols (e.g. starting with `L')
-M,--mri              assemble in MRI compatibility mode
--MD FILE             write dependency information in FILE (default none)
-nocpp                ignored
-o OBJFILE            name the object-file output OBJFILE (default a.out)
-R                    fold data section into text section
--reduce-memory-overheads
                        prefer smaller memory use at the cost of longer assembly
times
--statistics          print various measured statistics from execution
--strip-local-absolute strip local absolute symbols
--traditional-format  Use same format as native assembler when possible
--version             print assembler version number and exit
-W --no-warn          suppress warnings
--warn               don't suppress warnings
--fatal-warnings     treat warnings as errors
-w                   ignored
-X                   ignored
-Z                   generate object file even after errors
--listing-lhs-width  set the width in words of the output data column of the
listing
--listing-lhs-width2 set the width in words of the continuation lines of the
output data column; ignored if smaller than the width of the first line
--listing-rhs-width  set the max width in characters of the lines from the
source file
--listing-cont-lines set the maximum number of continuation lines used for the
output data column of the listing
@FILE                read options from FILE
AVR Assembler options:
-mmcu=[avr-name] select microcontroller variant
                        [avr-name] can be:
                        avr1 - classic AVR core without data RAM
                        avr2 - classic AVR core with up to 8K program memory
                        avr25 - classic AVR core with up to 8K program memory plus the
MOVW instruction
                        avr3 - classic AVR core with up to 64K program memory
                        avr31 - classic AVR core with up to 128K program memory
                        avr35 - classic AVR core with up to 64K program memory plus the
MOVW instruction
                        avr4 - enhanced AVR core with up to 8K program memory
                        avr5 - enhanced AVR core with up to 64K program memory
                        avr51 - enhanced AVR core with up to 128K program memory
                        avr6 - enhanced AVR core with up to 256K program memory
                        avrxmega2 - XMEGA, > 8K, < 64K FLASH, < 64K RAM
                        avrxmega3 - XMEGA, > 8K, <= 64K FLASH, > 64K RAM
                        avrxmega4 - XMEGA, > 64K, <= 128K FLASH, <= 64K RAM
                        avrxmega5 - XMEGA, > 64K, <= 128K FLASH, > 64K RAM
                        avrxmega6 - XMEGA, > 128K, <= 256K FLASH, <= 64K RAM
                        avrxmega7 - XMEGA, > 128K, <= 256K FLASH, > 64K RAM
                        avrtiny - AVR Tiny core with 16 gp registers
-mall-opcodes        accept all AVR opcodes, even if not supported by MCU
-mno-skip-bug        disable warnings for skipping two-word instructions (default for
avr4, avr5)
-mno-wrap            reject rjmp/rcall instructions with 8K wrap-around (default for
avr3, avr5)
-mrmw                accept Read-Modify-Write instructions
-mlink-relax         generate relocations for linker relaxation (default)
-mno-link-relax     don't generate relocations for linker relaxation.
-mnon-bit-addressable-registers-mask=<32 bit mask>
                        Specify non bit addressable registers mask.
Known MCU names:
avr1 avr2 avr25 avr3 avr31 avr35 avr4 avr5 avr51 avr6 avrxmega1

```

avrxmega2 avrxmega3 avrxmega4 avrxmega5 avrxmega6 avrxmega7 avrtiny  
at90s1200 attiny11 attiny12 attiny15 attiny28 at90s2313 at90s2323  
at90s2333 at90s2343 attiny22 attiny26 at90s4414 at90s4433 at90s4434  
at90s8515 at90c8534 at90s8535 ata5272 attiny13 attiny13a attiny2313  
attiny2313a attiny24 attiny24a attiny4313 attiny44 attiny44a attiny84  
attiny84a attiny25 attiny45 attiny85 attiny261 attiny261a attiny461  
attiny461a attiny861 attiny861a attiny87 attiny43u attiny48 attiny88  
attiny828 at86rf401 at43usb355 at76c711 atmega103 at43usb320 attiny167  
at90usb82 at90usb162 ata5505 atmega8u2 atmega16u2 atmega32u2 attiny1634  
atmega8 atmega8a ata6285 ata6286 ata6289 atmega48 atmega48a atmega48pa  
atmega48p atmega88 atmega88a atmega88p atmega88pa atmega8515 atmega8535  
atmega8hva at90pwm1 at90pwm2 at90pwm2b at90pwm3 at90pwm3b at90pwm81  
at90pwm161 ata5790 ata5795 atmega16 atmega16a atmega161 atmega162  
atmega163 atmega164a atmega164p atmega164pa atmega165 atmega165a  
atmega165p atmega165pa atmega168 atmega168a atmega168p atmega168pa  
atmega169 atmega169a atmega169p atmega169pa atmega32 atmega32a atmega323  
atmega324a atmega324p atmega324pa atmega325 atmega325a atmega325p  
atmega325pa atmega3250 atmega3250a atmega3250p atmega3250pa atmega328  
atmega328p atmega329 atmega329a atmega329p atmega329pa atmega3290  
atmega3290a atmega3290p atmega3290pa atmega406 atmega64rfr2  
atmega644rfr2 atmega64 atmega64a atmega640 atmega644 atmega644a  
atmega644p atmega644pa atmega645 atmega645a atmega645p atmega649  
atmega649a atmega649p atmega6450 atmega6450a atmega6450p atmega6490  
atmega6490a atmega6490p atmega64rfr2 atmega644rfr2 atmega16hva  
atmega16hva2 atmega16hvb atmega16hvbrevb atmega32hvb atmega32hvbrevb  
atmega64hve at90can32 at90can64 at90pwm161 at90pwm216 at90pwm316  
atmega32c1 atmega64c1 atmega16m1 atmega32m1 atmega64m1 atmega16u4  
atmega32u4 atmega32u6 at90usb646 at90usb647 at90scr100 at94k m3000  
atmega128 atmega128a atmega1280 atmega1281 atmega1284 atmega1284p  
atmega128rfal atmega128rfr2 atmega1284rfr2 at90can128 at90usb1286  
at90usb1287 atmega2560 atmega2561 atmega256rfr2 atmega2564rfr2  
atxmegal6a4 atxmegal6a4u atxmegal6c4 atxmegal6d4 atxmega32a4  
atxmega32a4u atxmega32c4 atxmega32d4 atxmega32e5 atxmegal6e5 atxmega8e5  
atxmega32x1 atxmega64a3 atxmega64a3u atxmega64a4u atxmega64b1  
atxmega64b3 atxmega64c3 atxmega64d3 atxmega64d4 atxmega64a1 atxmega64a1u  
atxmegal28a3 atxmegal28a3u atxmegal28b1 atxmegal28b3 atxmegal28c3  
atxmegal28d3 atxmegal28d4 atxmegal92a3 atxmegal92a3u atxmegal92c3  
atxmegal92d3 atxmega256a3 atxmega256a3u atxmega256a3b atxmega256a3bu  
atxmega256c3 atxmega256d3 atxmega384c3 atxmega384d3 atxmegal28a1  
atxmegal28a1u atxmegal28a4u attiny4 attiny5 attiny9 attiny10 attiny20  
attiny40

Report bugs to <<http://www.sourceware.org/bugzilla/>>

## 5.9 ( ) ld.exe

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\ld.exe

ld.exe 1 258 496 05.12.2016 17:55

### >ld.exe --help

Usage: ld [options] file...

Options:

-a KEYWORD	Shared library control for HP/UX compatibility
-A ARCH, --architecture ARCH	
	Set architecture
-b TARGET, --format TARGET	Specify target for following input files
-c FILE, --mri-script FILE	Read MRI format linker script

```

-d, -dc, -dp                Force common symbols to be defined
-e ADDRESS, --entry ADDRESS Set start address
-E, --export-dynamic        Export all dynamic symbols
--no-export-dynamic         Undo the effect of --export-dynamic
-EB                          Link big-endian objects
-EL                          Link little-endian objects
-f SHLIB, --auxiliary SHLIB Auxiliary filter for shared object symbol table
-F SHLIB, --filter SHLIB    Filter for shared object symbol table
-g                            Ignored
-G SIZE, --gpsize SIZE      Small data size (if no size, same as --shared)
-h FILENAME, -soname FILENAME
                              Set internal name of shared library
-I PROGRAM, --dynamic-linker PROGRAM
                              Set PROGRAM as the dynamic linker to use
--no-dynamic-linker         Produce an executable with no program interpreter
header
-l LIBNAME, --library LIBNAME
                              Search for library LIBNAME
-L DIRECTORY, --library-path DIRECTORY
                              Add DIRECTORY to library search path
--sysroot=<DIRECTORY>       Override the default sysroot location
-m EMULATION                 Set emulation
-M, --print-map              Print map file on standard output
-n, --nmagic                 Do not page align data
-N, --omagic                 Do not page align data, do not make text readonly
--no-omagic                 Page align data, make text readonly
-o FILE, --output FILE       Set output file name
-O                            Optimize output file
-plugin PLUGIN               Load named plugin
-plugin-opt ARG              Send arg to last-loaded plugin
-flto                        Ignored for GCC LTO option compatibility
-flto-partition=             Ignored for GCC LTO option compatibility
-fuse-ld=                    Ignored for GCC linker option compatibility
--map-whole-files            Ignored for gold option compatibility
--no-map-whole-files         Ignored for gold option compatibility
-Qy                           Ignored for SVR4 compatibility
-q, --emit-relocs            Generate relocations in final output
-r, -i, --relocatable        Generate relocatable output
-R FILE, --just-symbols FILE
                              Just link symbols (if directory, same as --rpath)
-s, --strip-all             Strip all symbols
-S, --strip-debug            Strip debugging symbols
--strip-discarded            Strip symbols in discarded sections
--no-strip-discarded         Do not strip symbols in discarded sections
-t, --trace                  Trace file opens
-T FILE, --script FILE       Read linker script
--default-script FILE, -dT   Read default linker script
-u SYMBOL, --undefined SYMBOL
                              Start with undefined reference to SYMBOL
--require-defined SYMBOL     Require SYMBOL be defined in the final output
--unique [=SECTION]          Don't merge input [SECTION | orphan] sections
-Ur                           Build global constructor/destructor tables
-v, --version                 Print version information
-V                             Print version and emulation information
-x, --discard-all           Discard all local symbols
-X, --discard-locals         Discard temporary local symbols (default)
--discard-none               Don't discard any local symbols
-y SYMBOL, --trace-symbol SYMBOL
                              Trace mentions of SYMBOL
-Y PATH                       Default search path for Solaris compatibility
-(, --start-group            Start a group
-), --end-group              End a group
--accept-unknown-input-arch Accept input files whose architecture cannot be de-
termined
--no-accept-unknown-input-arch
                              Reject input files whose architecture is unknown

```

```

--as-needed          Only set DT_NEEDED for following dynamic libs if used
--no-as-needed      Always set DT_NEEDED for dynamic libraries mentioned
on the command line
--assert KEYWORD    Ignored for SunOS compatibility
-Bdynamic, -dy, -call_shared
-Bstatic, -dn, -non_shared, -static
-Bsymbolic          Link against shared libraries
-Bsymbolic-functions Bind global references locally
--check-sections    Bind global function references locally
--no-check-sections Check section addresses for overlaps (default)
--copy-dt-needed-entries Do not check section addresses for overlaps
low
--no-copy-dt-needed-entries Copy DT_NEEDED links mentioned inside DSOs that follow
that follow
--cref              Output cross reference table
--defsym SYMBOL=EXPRESSION Define a symbol
--demangle [=STYLE] Demangle symbol names [using STYLE]
--embedded-relocs   Generate embedded relocs
--fatal-warnings     Treat warnings as errors
--no-fatal-warnings Do not treat warnings as errors (default)
--fini SYMBOL        Call SYMBOL at unload-time
--force-exe-suffix   Force generation of file with .exe suffix
--gc-sections        Remove unused sections (on some targets)
--no-gc-sections     Don't remove unused sections (default)
--print-gc-sections List removed unused sections on stderr
--no-print-gc-sections Do not list removed unused sections
--hash-size=<NUMBER> Set default hash table size close to <NUMBER>
--help              Print option help
--init SYMBOL        Call SYMBOL at load-time
-Map FILE           Write a map file
--no-define-common  Do not define Common storage
--no-demangle        Do not demangle symbol names
--no-keep-memory     Use less memory and more disk I/O
--no-undefined       Do not allow unresolved references in object files
--allow-shlib-undefined Allow unresolved references in shared libraries
--no-allow-shlib-undefined Do not allow unresolved references in shared libs
--allow-multiple-definition Allow multiple definitions
--no-undefined-version Disallow undefined version
--default-symver     Create default symbol version
--default-imported-symver Create default symbol version for imported symbols
--no-warn-mismatch   Don't warn about mismatched input files
--no-warn-search-mismatch Don't warn on finding an incompatible library
--no-whole-archive   Turn off --whole-archive
--noinhibit-exec    Create an output file even if errors occur
-nostdlib           Only use library directories specified on the command
line
--oformat TARGET     Specify target of output file
--print-output-format Print default output format
--print-sysroot      Print current sysroot
-qmagic             Ignored for Linux compatibility
--reduce-memory-overheads Reduce memory overheads, possibly taking much longer
--relax             Reduce code size by using target specific optimizations
--no-relax          Do not use relaxation techniques to reduce code size
--retain-symbols-file FILE Keep only symbols listed in FILE
-rpath PATH         Set runtime shared library search path
-rpath-link PATH    Set link time shared library search path
-shared, -Bshareable Create a shared library
-pie, --pic-executable Create a position independent executable
--sort-common [=ascending|descending]
Sort common symbols by alignment [in specified order]
--sort-section name|alignment
Sort sections by name or maximum alignment
--spare-dynamic-tags COUNT How many tags to reserve in .dynamic section

```



```

--split-by-file [=SIZE]      Split output sections every SIZE octets
--split-by-reloc [=COUNT]  Split output sections every COUNT relocs
--stats                      Print memory usage statistics
--target-help                Display target specific options
--task-link SYMBOL           Do task level linking
--traditional-format         Use same format as native linker
--section-start SECTION=ADDRESS
                             Set address of named section
-Tbss ADDRESS                Set address of .bss section
-Tdata ADDRESS               Set address of .data section
-Ttext ADDRESS               Set address of .text section
-Ttext-segment ADDRESS      Set address of text segment
-Trodata-segment ADDRESS    Set address of rodata segment
-Tldata-segment ADDRESS     Set address of ldata segment
--unresolved-symbols=<method>
                             How to handle unresolved symbols. <method> is:
                             ignore-all, report-all, ignore-in-object-files,
                             ignore-in-shared-libs
--verbose [=NUMBER]         Output lots of information during link
--version-script FILE        Read version information script
--version-exports-section SYMBOL
                             Take export symbols list from .exports, using SYMBOL
as the version.
--dynamic-list-data          Add data symbols to dynamic list
--dynamic-list-cpp-new      Use C++ operator new/delete dynamic list
--dynamic-list-cpp-typeinfo Use C++ typeid dynamic list
--dynamic-list FILE          Read dynamic list
--warn-common                Warn about duplicate common symbols
--warn-constructors          Warn if global constructors/destructors are seen
--warn-multiple-gp           Warn if the multiple GP values are used
--warn-once                  Warn only once per undefined symbol
--warn-section-align         Warn if start of section changes due to alignment
--warn-shared-textrel        Warn if shared object has DT_TEXTREL
--warn-alternate-em          Warn if an object has alternate ELF machine code
--warn-unresolved-symbols    Report unresolved symbols as warnings
--error-unresolved-symbols   Report unresolved symbols as errors
--whole-archive              Include all objects from following archives
--wrap SYMBOL                Use wrapper functions for SYMBOL
--ignore-unresolved-symbol SYMBOL
                             Unresolved SYMBOL will not cause an error or warning
--push-state                 Push state of flags governing input file handling
--pop-state                  Pop state of flags governing input file handling
--print-memory-usage         Report target memory usage
--orphan-handling =MODE     Control how orphan sections are handled.
@FILE                        Read options from FILE
ld: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec ver-
ilog tekhex binary ihex
ld: supported emulations: avr2 avr1 avr25 avr3 avr31 avr35 avr4 avr5 avr51 avr6
avr8mega1 avr8mega2 avr8mega3 avr8mega4 avr8mega5 avr8mega6 avr8mega7 avrtiny
ld: emulation specific options:
ELF emulations:
--build-id[=STYLE]          Generate build ID note
--compress-debug-sections=[none|zlib|zlib-gnu|zlib-gabi]
                             Compress DWARF debug sections using zlib
                             Default: none
-z common-page-size=SIZE    Set common page size to SIZE
-z max-page-size=SIZE        Set maximum page size to SIZE
-z defs                      Report unresolved symbols in object files.
-z muldefs                   Allow multiple definitions
-z execstack                 Mark executable as requiring executable stack
-z noexecstack               Mark executable as not requiring executable stack
avr2:
--pmem-wrap-around=<val>    Make the linker relaxation machine assume that a
                             program counter wrap-around occurs at address
                             <val>. Supported values: 8k, 16k, 32k and 64k.
--no-call-ret-replacement   The relaxation machine normally will

```

substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>` Specify the non bit addressable registers mask.

avr1:

`--pmem-wrap-around=<val>` Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.

`--no-call-ret-replacement` The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>` Specify the non bit addressable registers mask.

avr25:

`--pmem-wrap-around=<val>` Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.

`--no-call-ret-replacement` The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>` Specify the non bit addressable registers mask.

avr3:

`--pmem-wrap-around=<val>` Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.

`--no-call-ret-replacement` The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>`

Specify the non bit addressable registers mask.

avr31:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.
- debug-stubs Used for debugging avr-ld.
- debug-relax Used for debugging avr-ld.
- detailed-mem-usage Dump detailed memory usage (object file wise) to map file.
- non-bit-addressable-registers-mask=<32 bit mask> Specify the non bit addressable registers mask.

avr35:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.
- debug-stubs Used for debugging avr-ld.
- debug-relax Used for debugging avr-ld.
- detailed-mem-usage Dump detailed memory usage (object file wise) to map file.
- non-bit-addressable-registers-mask=<32 bit mask> Specify the non bit addressable registers mask.

avr4:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.
- debug-stubs Used for debugging avr-ld.
- debug-relax Used for debugging avr-ld.
- detailed-mem-usage Dump detailed memory usage (object file wise) to map file.
- non-bit-addressable-registers-mask=<32 bit mask> Specify the non bit addressable registers mask.

avr5:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump

```

        stub. You can de-active this with this switch.
--debug-stubs          Used for debugging avr-ld.
--debug-relax         Used for debugging avr-ld.
--detailed-mem-usage  Dump detailed memory usage (object file wise) to map
file.
--non-bit-addressable-registers-mask=<32 bit mask>
                        Specify the non bit addressable registers mask.
avr51:
--pmem-wrap-around=<val>  Make the linker relaxation machine assume that a
                        program counter wrap-around occurs at address
                        <val>. Supported values: 8k, 16k, 32k and 64k.
--no-call-ret-replacement  The relaxation machine normally will
                        substitute two immediately following call/ret
                        instructions by a single jump instruction.
                        This option disables this optimization.
--no-stubs             If the linker detects to attempt to access
                        an instruction beyond 128k by a reloc that
                        is limited to 128k max, it inserts a jump
                        stub. You can de-active this with this switch.
--debug-stubs          Used for debugging avr-ld.
--debug-relax         Used for debugging avr-ld.
--detailed-mem-usage  Dump detailed memory usage (object file wise) to map
file.
--non-bit-addressable-registers-mask=<32 bit mask>
                        Specify the non bit addressable registers mask.
avr6:
--pmem-wrap-around=<val>  Make the linker relaxation machine assume that a
                        program counter wrap-around occurs at address
                        <val>. Supported values: 8k, 16k, 32k and 64k.
--no-call-ret-replacement  The relaxation machine normally will
                        substitute two immediately following call/ret
                        instructions by a single jump instruction.
                        This option disables this optimization.
--no-stubs             If the linker detects to attempt to access
                        an instruction beyond 128k by a reloc that
                        is limited to 128k max, it inserts a jump
                        stub. You can de-active this with this switch.
--debug-stubs          Used for debugging avr-ld.
--debug-relax         Used for debugging avr-ld.
--detailed-mem-usage  Dump detailed memory usage (object file wise) to map
file.
--non-bit-addressable-registers-mask=<32 bit mask>
                        Specify the non bit addressable registers mask.
avrxcmega1:
--pmem-wrap-around=<val>  Make the linker relaxation machine assume that a
                        program counter wrap-around occurs at address
                        <val>. Supported values: 8k, 16k, 32k and 64k.
--no-call-ret-replacement  The relaxation machine normally will
                        substitute two immediately following call/ret
                        instructions by a single jump instruction.
                        This option disables this optimization.
--no-stubs             If the linker detects to attempt to access
                        an instruction beyond 128k by a reloc that
                        is limited to 128k max, it inserts a jump
                        stub. You can de-active this with this switch.
--debug-stubs          Used for debugging avr-ld.
--debug-relax         Used for debugging avr-ld.
--detailed-mem-usage  Dump detailed memory usage (object file wise) to map
file.
--non-bit-addressable-registers-mask=<32 bit mask>
                        Specify the non bit addressable registers mask.
avrxcmega2:
--pmem-wrap-around=<val>  Make the linker relaxation machine assume that a
                        program counter wrap-around occurs at address
                        <val>. Supported values: 8k, 16k, 32k and 64k.
--no-call-ret-replacement  The relaxation machine normally will

```

substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>` Specify the non bit addressable registers mask.

**avrxcmega3:**

`--pmem-wrap-around=<val>` Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.

`--no-call-ret-replacement` The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>` Specify the non bit addressable registers mask.

**avrxcmega4:**

`--pmem-wrap-around=<val>` Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.

`--no-call-ret-replacement` The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>` Specify the non bit addressable registers mask.

**avrxcmega5:**

`--pmem-wrap-around=<val>` Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.

`--no-call-ret-replacement` The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.

`--no-stubs` If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.

`--debug-stubs` Used for debugging avr-ld.

`--debug-relax` Used for debugging avr-ld.

`--detailed-mem-usage` Dump detailed memory usage (object file wise) to map file.

`--non-bit-addressable-registers-mask=<32 bit mask>`

Specify the non bit addressable registers mask.

avrxcmega6:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.
- debug-stubs Used for debugging avr-ld.
- debug-relax Used for debugging avr-ld.
- detailed-mem-usage Dump detailed memory usage (object file wise) to map file.
- non-bit-addressable-registers-mask=<32 bit mask> Specify the non bit addressable registers mask.

avrxcmega7:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.
- debug-stubs Used for debugging avr-ld.
- debug-relax Used for debugging avr-ld.
- detailed-mem-usage Dump detailed memory usage (object file wise) to map file.
- non-bit-addressable-registers-mask=<32 bit mask> Specify the non bit addressable registers mask.

avrtiny:

- pmem-wrap-around=<val> Make the linker relaxation machine assume that a program counter wrap-around occurs at address <val>. Supported values: 8k, 16k, 32k and 64k.
- no-call-ret-replacement The relaxation machine normally will substitute two immediately following call/ret instructions by a single jump instruction. This option disables this optimization.
- no-stubs If the linker detects to attempt to access an instruction beyond 128k by a reloc that is limited to 128k max, it inserts a jump stub. You can de-active this with this switch.
- debug-stubs Used for debugging avr-ld.
- debug-relax Used for debugging avr-ld.
- detailed-mem-usage Dump detailed memory usage (object file wise) to map file.
- non-bit-addressable-registers-mask=<32 bit mask> Specify the non bit addressable registers mask.

Report bugs to <<http://www.sourceware.org/bugzilla/>>

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin\objdump.exe

objdump.exe 1 026 048 05.12.2016 17:53

d:\FOR\_STUDENTS\arduino-1.8.5\hardware\tools\avr\avr\bin>objdump --help

Usage: objdump <option(s)> <file(s)>

Display information from object <file(s)>.

At least one of the following switches must be given:

-a, --archive-headers	Display archive header information
-f, --file-headers	Display the contents of the overall file header
-p, --private-headers	Display object format specific file header contents
-P, --private=OPT,OPT...	Display object format specific contents
-h,--[section-]headers	Display the contents of the section headers
-x, --all-headers	Display the contents of all headers
-d, --disassemble	Display assembler contents of executable sections
-D, --disassemble-all	Display assembler contents of all sections
-S, --source	Intermix source code with disassembly
-s, --full-contents	Display the full contents of all sections requested
-g, --debugging	Display debug information in object file
-e, --debugging-tags	Display debug information using ctags style
-G, --stabs	Display (in raw form) any STABS info in the file
-W[llLiaprmmfFsoRt]	or
--dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=frames-interp,=str,=loc,=Ranges,=pubtypes,=gdb_index,=trace_info,=trace_abbrev,=trace_aranges,=addr,=cu_index]	
-t, --syms	Display DWARF info in the file
-T, --dynamic-syms	Display the contents of the symbol table(s)
-r, --reloc	Display the contents of the dynamic symbol table
-R, --dynamic-reloc	Display the relocation entries in the file
@<file>	Display the dynamic relocation entries in the file
-v, --version	Read options from <file>
-i, --info	Display this program's version number
-H, --help	List object formats and architectures supported
	Display this information

The following switches are optional:

-b, --target=BFDNAME	Specify the target object format as BFDNAME
-m, --architecture=MACHINE	Specify the target architecture as MACHINE
-j, --section=NAME	Only display information for section NAME
-M, --disassembler-options=OPT	Pass text OPT on to the disassembler
-EB --endian=big	Assume big endian format when disassembling
-EL --endian=little	Assume little endian format when disassembling
--file-start-context	Include context from start of file (with -S)
-I, --include=DIR	Add DIR to search list for source files
-l, --line-numbers	Include line numbers and filenames in output
-F, --file-offsets	Include file offsets when displaying information
-C, --demangle[=STYLE]	Decode mangled/processed symbol names
	The STYLE, if specified, can be `auto', `gnu', `lucid', `arm', `hp', `edg', `gnu-v3', `java' or `gnat'
-w, --wide	Format output for more than 80 columns
-z, --disassemble-zeroes	Do not skip blocks of zeroes when disassembling
--start-address=ADDR	Only process data whose address is >= ADDR
--stop-address=ADDR	Only process data whose address is <= ADDR
--prefix-addresses	Print complete address alongside disassembly
--[no-]show-raw-insn	Display hex alongside symbolic disassembly
--insn-width=WIDTH	Display WIDTH bytes on a single line for -d
--adjust-vma=OFFSET	Add OFFSET to all displayed section addresses

```

--special-syms          Include special symbols in symbol dumps
--prefix=PREFIX        Add PREFIX to absolute paths for -S
--prefix-strip=LEVEL   Strip initial directory names for -S
--dwarf-depth=N       Do not display DIEs at depth N or greater
--dwarf-start=N       Display DIEs starting with N, at the same depth or
deeper
--dwarf-check          Make additional dwarf internal consistency checks.

```

objdump: supported targets: elf32-avr elf32-little elf32-big plugin srec symbolsrec  
verilog tekhex binary ihex

objdump: supported architectures: avr avr:1 avr:2 avr:25 avr:3 avr:31 avr:35 avr:4  
avr:5 avr:51 avr:6 avr:100 avr:101 avr:102 avr:103 avr:104 avr:105 avr:106 avr:107  
plugin

Options supported for -P/--private switch:

For AVR ELF files:

```

mem-usage  Display memory usage
avr-prop   Display contents of .avr.prop section

```

Report bugs to <<http://www.sourceware.org/bugzilla/>>.

```

                                     GCC
                                     '
, NOP . ' -
                                     .
                                     .
                                     .
                                     ,
                                     .
                                     .
asm                                     ( ) -
asm.
asm - ,
                                     .
                                     .
                                     ,
                                     .
asm
asm( " "
:
:
:

```



```

:                                     );
                                     ".:",
                                     .
                                     .
                                     ,
volatile.                             asm
:
asm volatile ( . . .
. . . );

__asm__ __volatile__
AVR
:
__asm__ __volatile__ ( "cli" );
__asm__ __volatile__ ( "wdr" );
__asm__ __volatile__ ( "sei" );
( inline )

1)
,
asm volatile(
"clr r30 \n\t"
"clr r31 \n\t"
"ijmp \n\t"
);
2)
%0,
- %1 . .
%2.
3)
4)
( )
: $47 $0xA050. "$".
5)
"\ " s -

```

– "\n".  
(tabs).

```
asm volatile ( "ldi 16, %0" :: "i" (RAMEND >> 8) );  
asm volatile ( "out %0,16" :: "i" (AVR_STACK_POINTER_HI_ADDR) );  
asm volatile ( "ldi 16, %0" :: "i" (RAMEND & 0x0ff) );  
asm volatile ( "out %0,16" :: "i" (AVR_STACK_POINTER_LO_ADDR) );
```

RAM

1)

2)

(constraint) "r".

3)

```
"a" %%  
"b" %% bx
```

4)

(constraint) "m".

5)

(constraint) "=",

counter

```
asm ( "incw %0; "  
      : "= " (counter)  
      : "0" (counter) );
```

6)

7)

0 (n-1), r e n –

5,

%4.

(clobbered registers) –

```
asm( . . .  
    : "%esi", "%esi");
```

```
4  
__asm__ __volatile__ (  
    "nop" "\n\t"  
    "nop" "\n\t"  
    "nop" "\n\t"  
    "nop");
```

```
us:  
__asm__ __volatile__ (  
    "lb: sbiw %0,1" "\n\t" // 2  
    "brne lb" : "=w" (us) : "0" (us) // 2  
);
```

```
asm volatile ( "LSL %0 \n\t"  
    "BRCC 10 \n\t"  
    "INC %0 \n\t"  
    "l0: \n\t"  
    : "=r" (vLeds)  
    : "0" (vLeds));
```

### Arduino.h

```
#define interrupts() sei()  
#define noInterrupts() cli()  
#define _NOP() do { __asm__ volatile ("nop"); } while (0)
```

1. . . . :  
 . 1-31 04 02 « » , 1-31 04 03 « ».  
 .: , 2011. – 156 .