

generate binary code that will be burnt on board.

```
MINGW32://Desktop-77dmlj/f$/all/unit 3/lesson2/HW

--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       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
C:\ARM_TOOLCHAIN\bin\arm-none-eabi-objcopy.exe: supported targets: elf32-littlearm elf32-
bigarm elf32-little elf32-big srec symbolsrec verilog tekhex binary ihex

Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 //Desktop-77dmlj/f$/all/unit 3/lesson2/HW
$ arm-none-eabi-ld.exe -T linker.ld -Map=file.map main.o startup.o uart.o -o output.elf

Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 //Desktop-77dmlj/f$/all/unit 3/lesson2/HW
$ arm-none-eabi-objcopy.exe -O binary output.elf output.bin

Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 //Desktop-77dmlj/f$/all/unit 3/lesson2/HW
$ |
```

call qemo emulator to run the code on the board and see the expected output

```
MINGW32:/c/Program Files (x86)/qemu

Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 /c/Program Files (x86)/qemu
$ ./qemu-system-arm.exe -M versatilepb -m 128M -nographic -kernel output.bin
learn-in-depth:ELKOMY
Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 /c/Program Files (x86)/qemu
$ |
```

the symbols before and after linking

```
MINGW32://Desktop-77dmlj/f$/all/unit 3/lesson2/HW
Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 //Desktop-77dmlj/f$/all/unit 3/lesson2/HW
$ arm-none-eabi-objdump.exe -h main.o

main.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          00000018  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data          00000064  00000000  00000000  0000004c  2**2
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss           00000000  00000000  00000000  000000b0  2**0
    ALLOC
  3 .comment       00000012  00000000  00000000  000000b0  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 00000032  00000000  00000000  000000c2  2**0
    CONTENTS, READONLY

Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 //Desktop-77dmlj/f$/all/unit 3/lesson2/HW
$ arm-none-eabi-nm.exe main.o output.elf

main.o:
00000000 D arr
00000000 T main
          U uart_send_string

output.elf:
00010078 D arr
00010010 T main
00010000 T reset
000110dc D stack_top
00010008 t stop
00010028 T uart_send_string

Mohamed_ELkomy@DESKTOP-77DMLIJ MINGW32 //Desktop-77dmlj/f$/all/unit 3/lesson2/HW
```