



z88dk

The z80 development kit

[Home](#) [Downloads](#) [Docs](#) [Screenshots](#) [Targets](#) [Forum](#) [Source](#)

What is z88dk?

z88dk is the only C and assembler development kit that comes ready out-of-the-box to create programs for over 100 z80-family (8080, 8085, gbz80, z80, z180, ez80_z80, KC160, Rabbit 2000, 3000, 4000, 5000) machines.

The project was started in 1998/9 to allow a TCP stack for the Cambridge z88 to be easily written. In recent years significant work has been undertaken to improve the number of targets supported, enhance library features, make the toolchain easier to use and improve life for developers.

z88dk is dedicated to z80 family machines

Because we only target z80 family machines we can create optimised libraries. When you use z88dk, you automatically use a C standard library that is mostly written in assembler.

The [library](#) consists of over 15000 files containing at least 250k lines of assembler code. It's probably the largest repository of z80 assembler source code online.

z88dk is flexible

The point of any development kit should be to make life easy for you, the developer, and allow you to focus on writing your application and not on figuring out how to turn a hex file into a loadable artefact. As a result, the default setup for z88dk is configured to just make things work.

However, we know developers love twiddling knobs, after all, we're developers too. So with z88dk you can tune sizes, toggle features on and off, choose different floating point libraries, switch library implementations, override library implementations, bypass crt's, turn off using the standard library.

And yes, you can even generate hex files should you wish!

z88dk is easy to use

Assuming you want to compile for a supported target, and with over 100 targets to choose from, the chances are you are! Then it's easy to produce an "executable" with z88dk, it's just a single command. Let's create a .TAP file for use with ZX Spectrum emulators:

```
zcc +zx program.c -create-app
```

How about a bootable +3 Spectrum disc?

```
zcc +zx program.c -create-app -subtype=plus3
```

Need it for CP/M instead? Try this:

```
zcc +cpm program.c -create-app
```

And a `.com` file is created. Do you want to test the `.com` file using z80pack? It's much easier to test with a disc image, so let us create one:

```
zcc +cpm program.c -create-app -subtype=z80pack
```

Oh, it's an 8080 CP/M machine you're targetting, of course:

```
zcc +cpm -clib=8080 program.c -create-app -subtype=z80pack
```

An 8080 CP/M machine with a Polymorphic VTI graphics card?

```
zcc +cpm -clib=8080 program.c -create-app -subtype=z80pack --vti
```

z88dk is the easiest way to use sdcc for z80

Seriously it really is, lets do it:

```
zcc +cpm program.c -create-app -subtype=z80pack -compiler=sdcc
```

Easy as that, no messing around with crt files, linker scripts, hex2bin, cpmtools etc, just one command producing a disk image containing a binary compiled with sdcc that takes advantage of our optimised libraries.

z88dk is also the easiest way to use llvm for the z80

We've added experimental support for the ez80-clang build:

```
zcc +cpm program.c -create-app -subtype=z80pack -compiler=ez80clang
```

When using `-compiler=ez80clang` you can even write C++ (the standard library isn't available yet).

