# DAI vintage computer C graphical interfaces

# using z88dk C compiler

# Overview

This project details how to use the Z88DK C compiler ([*github.com/z88dk/z88dk/wiki/Platform---Dai*](https://github.com/z88dk/z88dk/wiki/Platform---Dai)*)* with a DAI, a vintage computer from Data Application International. It also provides C interfaces to call ROM graphical functions (not available for the time being on the z88dk).

# Z88DK for DAI

The Z88DK is a complete development toolkit for the 8080, 8085, gbz80, z80, z180, z80n, kc160 and Rabbit processors. It contains two C compilers, an assembler / linker / librarian, data compression tools and a utility for processing the raw binaries into forms needed by specific targets. Development in assembly language or C is completely integrated; projects can be 100% assembler, 100% C or any mixture of the two.

Support for the DAI is only partial:

* Limited to the classic library. Balloc allocation, algorithm/adt extensions and debug libraries are not available
* Graphics extensions are not available, but all DAI graphical functions can be used using interfaces to the DAI ROM (see attached examples)

The z88dk classic library supports compiler intrinsic (l\_) functions optimized for 8080, 8085, gbz80, or z80 CPUs. There are two [maths libraries](https://github.com/z88dk/z88dk/wiki/Classic--Maths-Libraries) that are available for the 8080/8085: dai32, extracted from the [DAI](https://github.com/z88dk/z88dk/wiki/Platform---Dai) ROM, and mbf32 Microsoft Basic Floating Point, extracted from Microsoft Basic 4.7. DAI coprocessor AMD9511 is supported, see:

[github.com/z88dk/z88dk/tree/master/libsrc/math/am9511](https://github.com/z88dk/z88dk/tree/master/libsrc/math/am9511)

# Z88DK installation

Download source / executable of nightly versions: [*github.com/z88dk/z88dk-build*](https://github.com/z88dk/z88dk-build)*.* Do not use stable release version 2.3 because a patch related to .wav files has been introduced since February 19th, 2024.

Installation: github.com/z88dk/z88dk/wiki/installation. For windows, one user variable and a system path must be set. As an alternative to Cmake Visual studio solution can be used to compile the source (open the z88dk.sln in win32 directory). Dll libraries can be copied from lib /clibs directory of the stable release version 2.3. Virustotal.com suggests that there are potential virus issues (github.com/z88dk/z88dk/issues/2474).

# Compiling and executing programs

Assuming “a.c” is the program to be compiled for the DAI.

Command line to compile the program and to generate wav and log files:

* zcc.exe +dai -m -v -s -create-app --list --math-mbf32 -Cz--loud a.c >> a.txt
* zcc.exe +dai -m -v -s -create-app --list --math-dai32 -Cz--loud a.c >> a.txt

Library --math-dai32 correspond to math library from the DAI Rom, whereas --math-mbf32 comes from Microsoft (required for the Mandelbrot example)

For DAI version V4/V5, the -Cz--loud option, is recommended to generates a wav file with the loudest volume.

To get assembly program: z88dk-dis.exe -m8080 -o 0x0800 -x a.map a.bin >> a.asm

# Executing compiled programs on a DAI

To execute the program “a.c”, load the generated “a.wav” file. From Basic, type:

* UT (to go to Utility)
* Z3 (to reset stack, not compulsory)
* R (to read the file)
* G0800 (to run the program)

The program will exit with a Stack overflow error when using MAME but not with a real DAI.

# Programming specificities

Programming in C on the DAI leads to some constraints, notably linked to stack size (128 bytes from $F800 to $F8FF). To limits problems:

* Use short strings / insert carriage returns as soon as possible (“\n”) with printf function
* Do not use too frequently printf function and insert delay loops if necessary (interruptions need time to process characters outputs)
* Use static variables to limit stack usage
* Pay attention to the chosen math library. Floats are converted in double (32bits) but the Dai32 library is problematic for small numbers (mantissa of only 7bits)
* Use unambiguous type definitions (ex: int32\_t from stdint.h) when using existing C programs, because for example an int on a windows machine is 4 bytes, but 2 bytes on the z88dk compiler

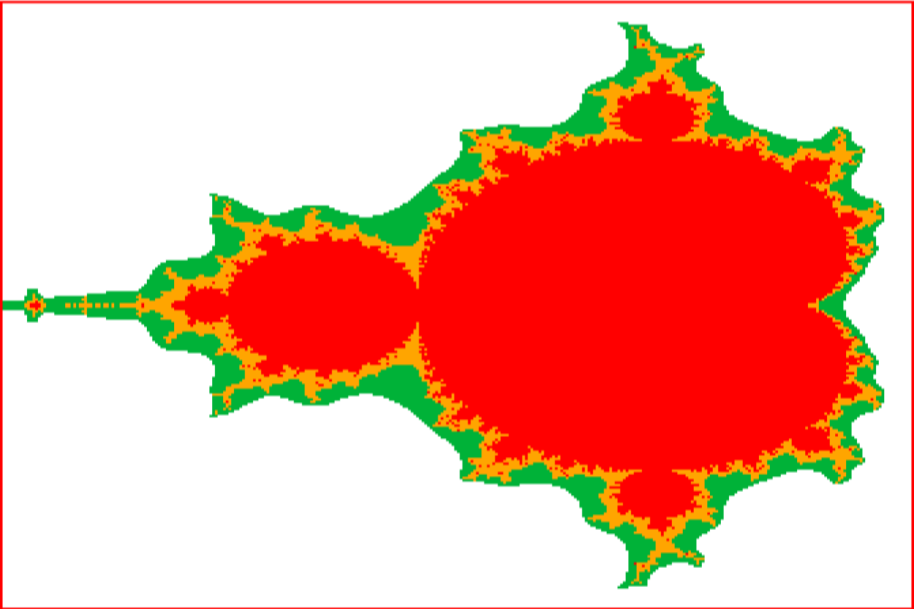
Attached file contains a C example to change the position of the stack if necessary.

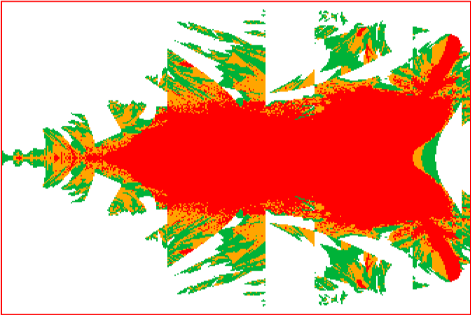
## Graphical example

The Z88dk Mandelbrot fractal example has been modified to run on a DAI. See attached c file for additional examples.

Main modifications are:

* C interfaces to call DAI graphical functions
* Use of static variables to avoid stack overflow
* Use of mbf32 math library because dai32 library is not precise enough as shown below





Version with DAI32: precision in not sufficient