

A [new release](#) of SDCC, the portable optimizing compiler for 8051, DS390, Z80, Z180, Rabbit 2000, GBZ80, eZ80 in Z80 mode, TLCS-90,

HC08, STM8, Padauk and PIC microprocessors is now available. Sources, documentation and binaries for GNU/Linux amd64, Windows amd64, macOS amd64 and Windows x86 are available.

What is SDCC?

SDCC is a retargettable, optimizing Standard C (ANSI C89, ISO C99, ISO C11) compiler suite that targets the Intel MCS51 based microprocessors (8031, 8032, 8051, 8052, etc.), Maxim (formerly Dallas) DS80C390 variants, Freescale (formerly Motorola) HC08 based (hc08, s08), Zilog Z80 based MCUs (Z80, Z180, SM83, Rabbit 2000, 2000A, 3000A, TLCS-90), Padauk (pdk14, pdk15) and STMicroelectronics STM8. Work is in progress on supporting the Padauk pdk13 and MOS 6502 targets; Microchip PIC16 and PIC18 targets are unmaintained. SDCC can be retargeted for other microprocessors.

SDCC suite is a collection of several components derived from different sources with different FOSS licenses. SDCC compiler suite include:

- sdas and sldd, a retargettable assembler and linker, based on ASXXXX, written by Alan Baldwin; (GPL).
- sdcpp preprocessor, based on GCC cpp; (GPL).
- ucsim simulators, originally written by Daniel Drotos; (GPL).
- sdcdb source level debugger, originally written by Sandeep Dutta; (GPL).
- sdbinutils library archive utilities, including sdar, sdranlib and sdnm, derived from GNU Binutils; (GPL)
- SDCC run-time libraries; (GPL+LE). Pic device libraries and header files are derived from Microchip header (.inc) and linker script (.lkr) files. Microchip requires that "The header files should state that they are only to be used with authentic Microchip devices" which makes them incompatible with the GPL.
- gcc-test regression tests, derived from gcc-testsuite; (no license explicitly specified, but since it is a part of GCC is probably GPL licensed)
- packihx; (public domain)
- makebin; (zlib/libpng License)
- sdcc C compiler, originally written by Sandeep Dutta; (GPL). Some of the features include:
 - extensive MCU specific language extensions, allowing effective use of the underlying hardware.
 - a host of standard optimizations such as global sub expression elimination, loop optimizations (loop invariant, strength reduction of induction variables and loop reversing), constant folding and propagation, copy propagation, dead code elimination and jump tables for 'switch' statements.
 - MCU specific optimizations, including a global register allocator.
 - adaptable MCU specific backend that should be well suited for other 8 bit MCUs
 - independent rule based peep hole optimizer.
 - a full range of data types: char (8 bits, 1 byte), short (16 bits, 2 bytes), int (16 bits, 2 bytes), long (32 bit, 4 bytes), long long (64 bit, 8 bytes), float (4 byte IEEE) and _Bool/bool.
 - the ability to add inline assembler code anywhere in a function.
 - the ability to report on the complexity of a function to help decide what should be re-written in assembler.
 - a good selection of automated regression tests.

SDCC was originally written by Sandeep Dutta and released under a GPL license. Since its initial release there have been numerous bug fixes and improvements. As of December 1999, the code was moved to SourceForge where all the "users turned developers" can access the same source tree. SDCC is constantly being updated with all the users' and developers' input.

SDCC 4.2.0 New Feature List:

- C23 memset_explicit
- Support for --oldralloc has been removed from the z80, z180, tlcs90, z80n, ez80_z80, r2k, r2ka, r3ka backends.
- gbz80 port now uses more efficient block-initialization of global variables (users of a custom crt0 need to adapt theirs).
- Full support for __z88dk_callee for the z80, z180, gbz80, tlcs90, z80n, ez80_z80, r2k, r2ka, r3ka, stm8 backends.
- Support for __raisonance, __iar and __cosmic calling conventions for stm8.
- Support for a new __sdcccall(1) calling convention in the stm8 port AS NEW DEFAULT.
- Support for a new __sdcccall(1) calling convention in the gbz80 port AS NEW DEFAULT.
- Support for a new __sdcccall(1) calling convention in the z80, z80n and z180 ports AS NEW DEFAULT.
- Support for a new __sdcccall(1) calling convention in the r2k, r2ka, r3k, tlcs90 and ez80_z80 ports.
- Removed support for --profile for gbz80, z80, z180, tlcs90, z80n, ez80_z80, r2k, r2ka, r3ka backends.
- The z80n port Z80N Core minimum version has been raised from 1.0 to 2.0.
- Improved rematerialization support in the stm8, gbz80, z80, z180, tlcs90, z80n, ez80_z80, r2k, r2ka, r3ka backends.
- The gbz80 port was renamed to sm83.
- New in-development mos6502 port.

Numerous other new features and bug fixes are included as well.

You can download the release from: <https://sourceforge.net/projects/sdcc/files/>