Developing Firmware and Flashing Updates to CH559 MCU Windows Platform Tools Installation Tutorial

Latest update: March 14th 2023

The Tools Necessary

- A code editor aka IDE – we'll install VS Code
- A BASH shell – we'll obtain Git Bash by installing Git for Windows
- The Small Device C Compiler aka SDCC
- WCHISPStudio chip flash tool from manufacturer NanJing Qinheng Microelectronics

note: the following instructions were tested on a 64 bit Windows 10 installation. Bear that in mind when visiting the download pages. I've not tested, but I do suspect the flash tool (the last program to download) may not work on 32 bit Windows

Installing VS Code

- Open up a web browser and visit https://code.visualstudio.com/
- Click the "Download for Windows" button link.
- When the download finishes, execute the setup program.
- Accept License Terms and click Next.
- At "Select Destination Location" accept the path suggested within your user profile directory or change it to another of your choosing.
- At "Select Start Menu Folder" click Next.
- At "Select Additional Tasks" mark your setting preferences.

  - Note: the Add "Open with Code" context menu options are handy to have set.
- Click Next.
- Click Install, after install is done, click Finish.

Installing Git for Windows

- Navigate browser to https://git-scm.com/download/win
- Click the "Click here to download" the latest version link.
- When the download finishes, execute the setup program.
- Click Next at License dialog.
- At "Select Destination Location" dialog, pick a path and click Next.

- At “Select Components” check boxes,
  add a desktop icon if you like. All other default values are fine.
- At “Select Start Menu Folder”, click Next.
- At “Choosing the default editor used by Git”, change the dropdown to
  “Use Visual Studio Code as Git’s default editor”, click Next.
- Adjust the initial branch naming practice if you like. Click Next.
- For “Adjusting your PATH environment”,
  the middle option “Git from command line…” will do fine. Click Next.
- “Use bundled OpenSSH”, click Next. “Use the OpenSSL library”, click Next.
- “Configuring the line ending conversions” should stay on top/default value. Click Next.
- “Configuring the terminal emulator to use with Git Bash” should stay with “MinTTY”.
- Click Next. Leave “behavior of git pull” at the default value. Click Next.
- For purposes of this tutorial “Choose credential helper” setting won’t matter.
  So, whichever you like, “Git Credential Manager” is the recommended setting, click Next.
- “Configuring extra options” settings are good by default, click Next.
- “Configuring experimental options” can all stay unchecked. Click Install. Click Finish.

Installing SDCC

- Navigate browser to https://sdcc.sourceforge.net/ to read a bit about it.
- And to https://sdcc.sourceforge.net/snap.php to download a binary snapshot.
- Scroll down to the Windows 64 bit setup programs and click link for latest setup exe.

Supported Windows - x86_64 Binaries

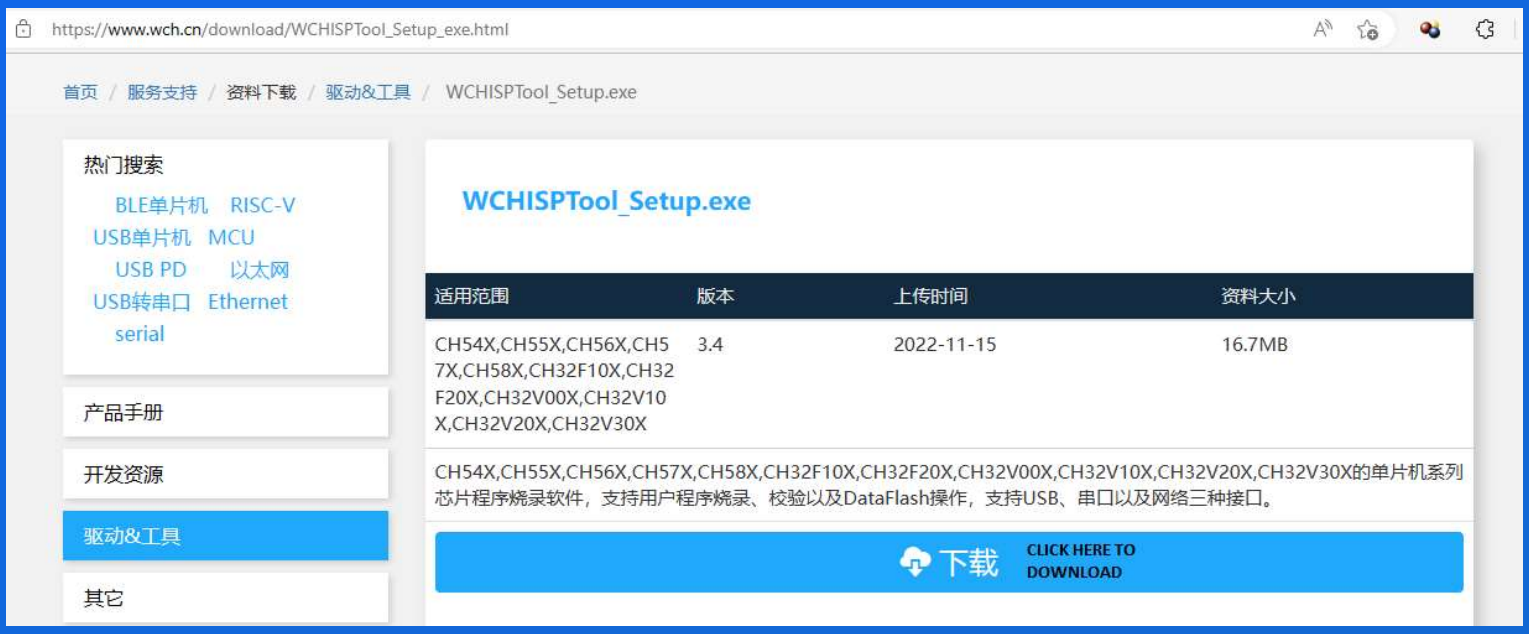
Windows 64bit package with installer (x86_64-w64-mingw32-setup)

Filename	Size	Snapshot Date
sdcc-20230312-13904-x64-setup.exe	7603 K	Mar 12 12:00
sdcc-20230311-13904-x64-setup.exe	7596 K	Mar 11 11:45
sdcc-20230310-13904-x64-setup.exe	7604 K	Mar 10 11:30
sdcc-20230309-13904-x64-setup.exe	7601 K	Mar 9 11:30
sdcc-20230308-13902-x64-setup.exe	7603 K	Mar 8 11:45

- When the download finishes, execute the setup program.
- At “Welcome to SDCC 4.x.xx Setup”, welcome form, click Next.
- At “License Agreement”, click I Agree button, then Next for Start Menu Folder.
- At “Choose Components”, click Next.

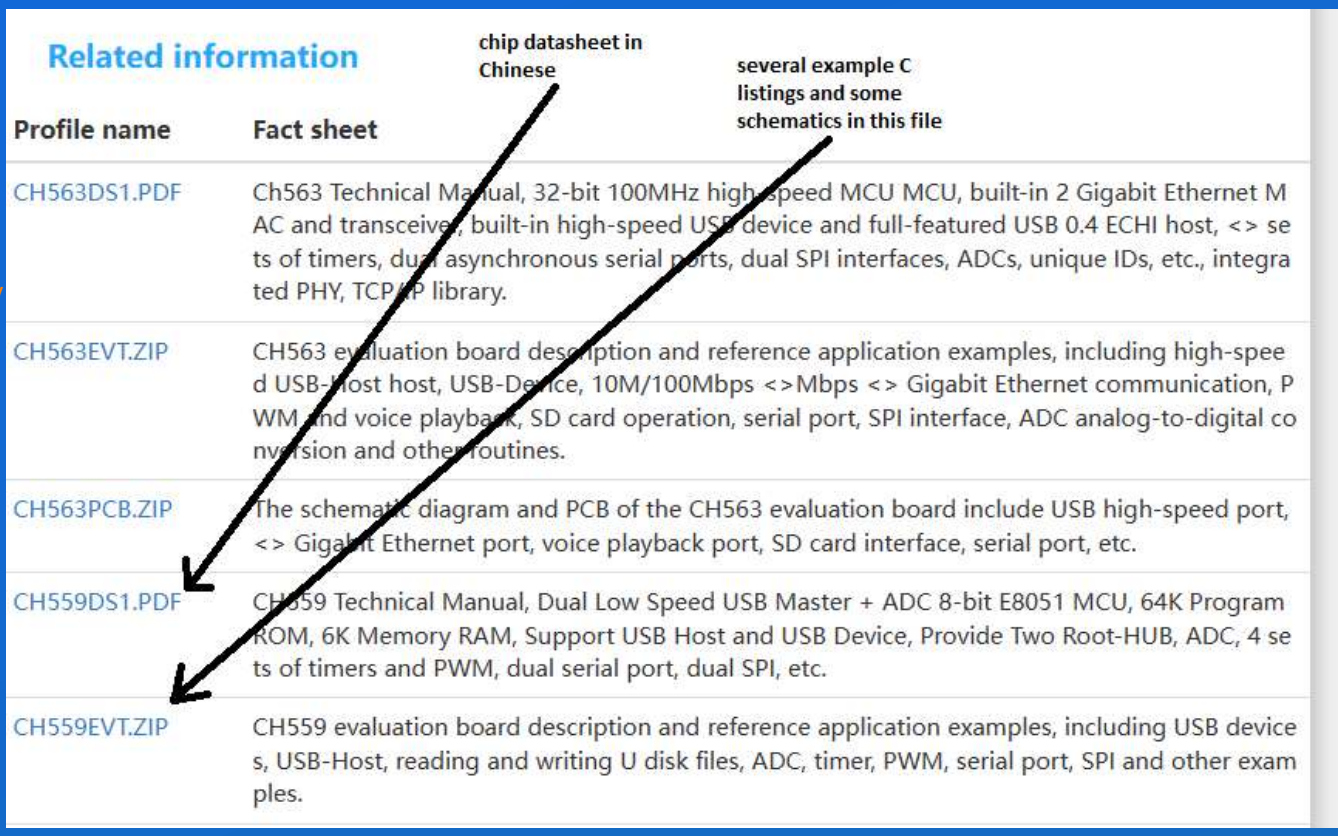
- At “Completing SDCC Setup”, make sure Add to the PATH is checked. Click Next.
- Click Finish.

The WCHISPTool USB Flash Tool

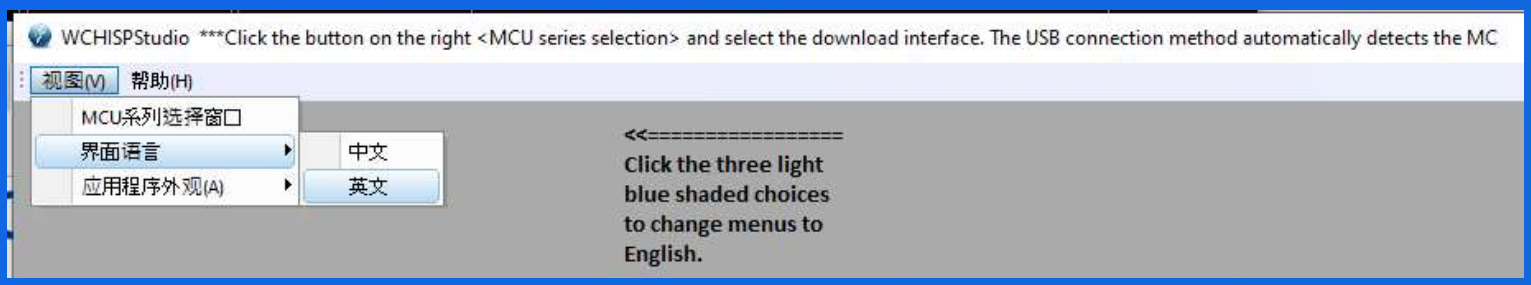


- Navigate browser to https://www.wch.cn/download/WCHISPTool_Setup_exe.html
- Use big blue Download button(pictured above), then run installer program.

The two indicated files are supplemental information you may download for reference.

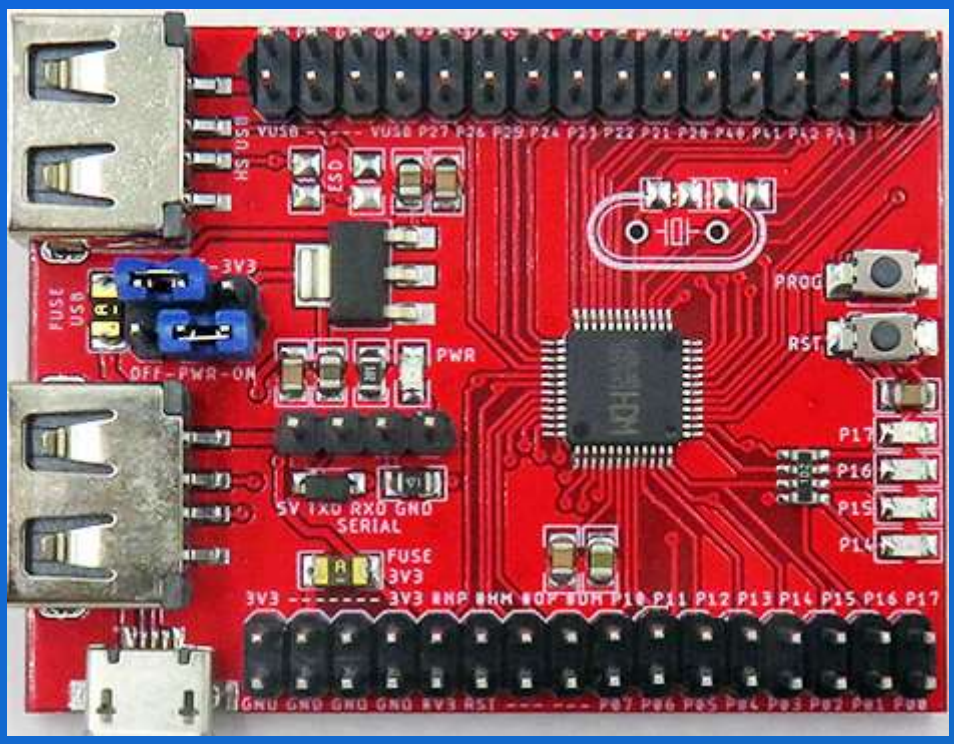


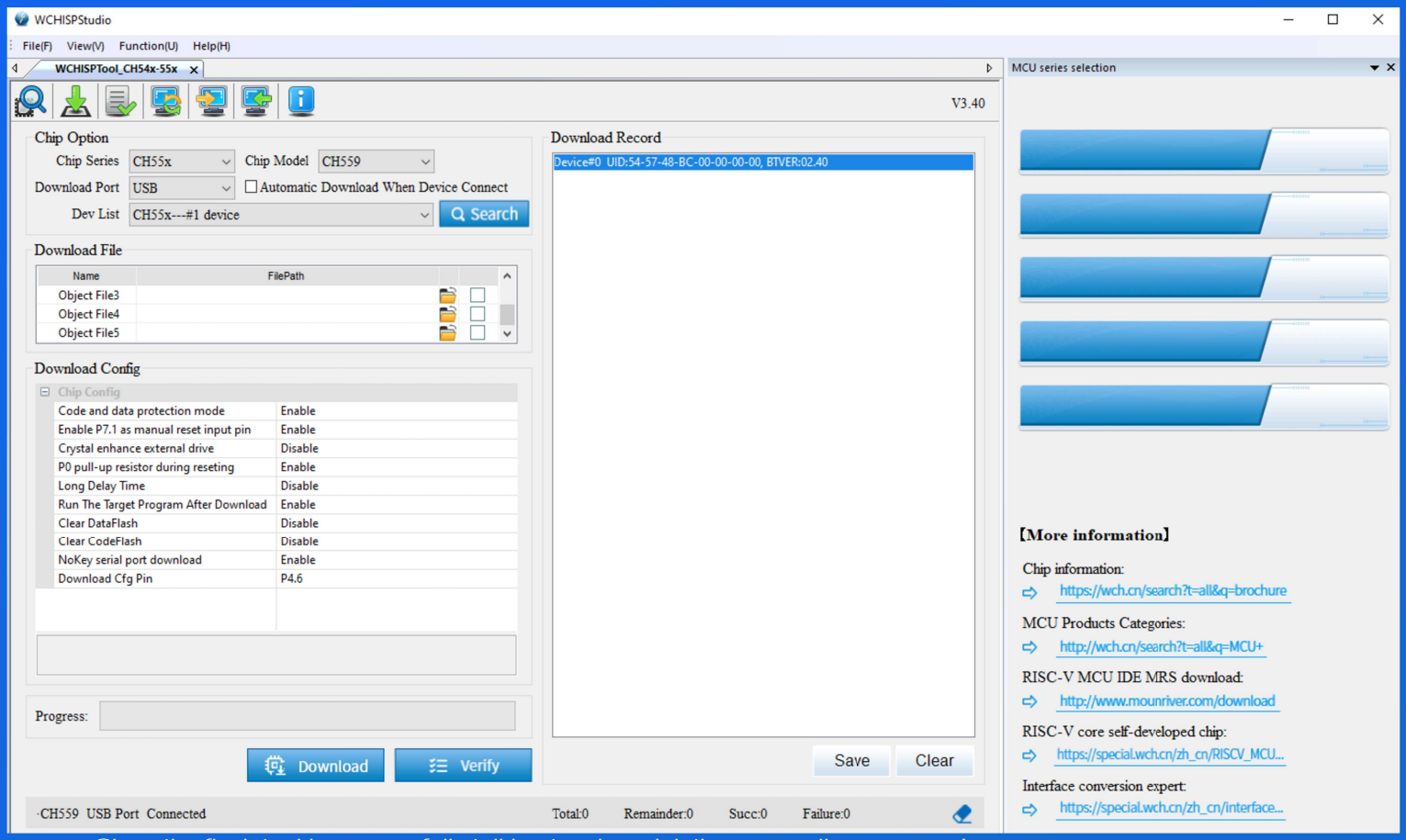
- At the Welcome to WCHISPTool Setup form, click Next.
- At “License Agreement” form, click I Agree button.
- At “Choose Install Location” form, leave default path or enter a preference, click Next.
- At “Choose Start Menu Folder” form, click Install button. Click Finish. Start application.



- The program starts with a large gray empty project area. (now, time to attach dev board)
- While holding down the (PROG)program button, attach a USB cable from your computer to the CH559 development board at the side micro USB port.

When the introduced USB device is recognized by the computer, release the PROG button.
The flash tool application should detect the board and fill that empty project pane as on next page. It will likely also remember the project and from now on display the project even when no board is detected.





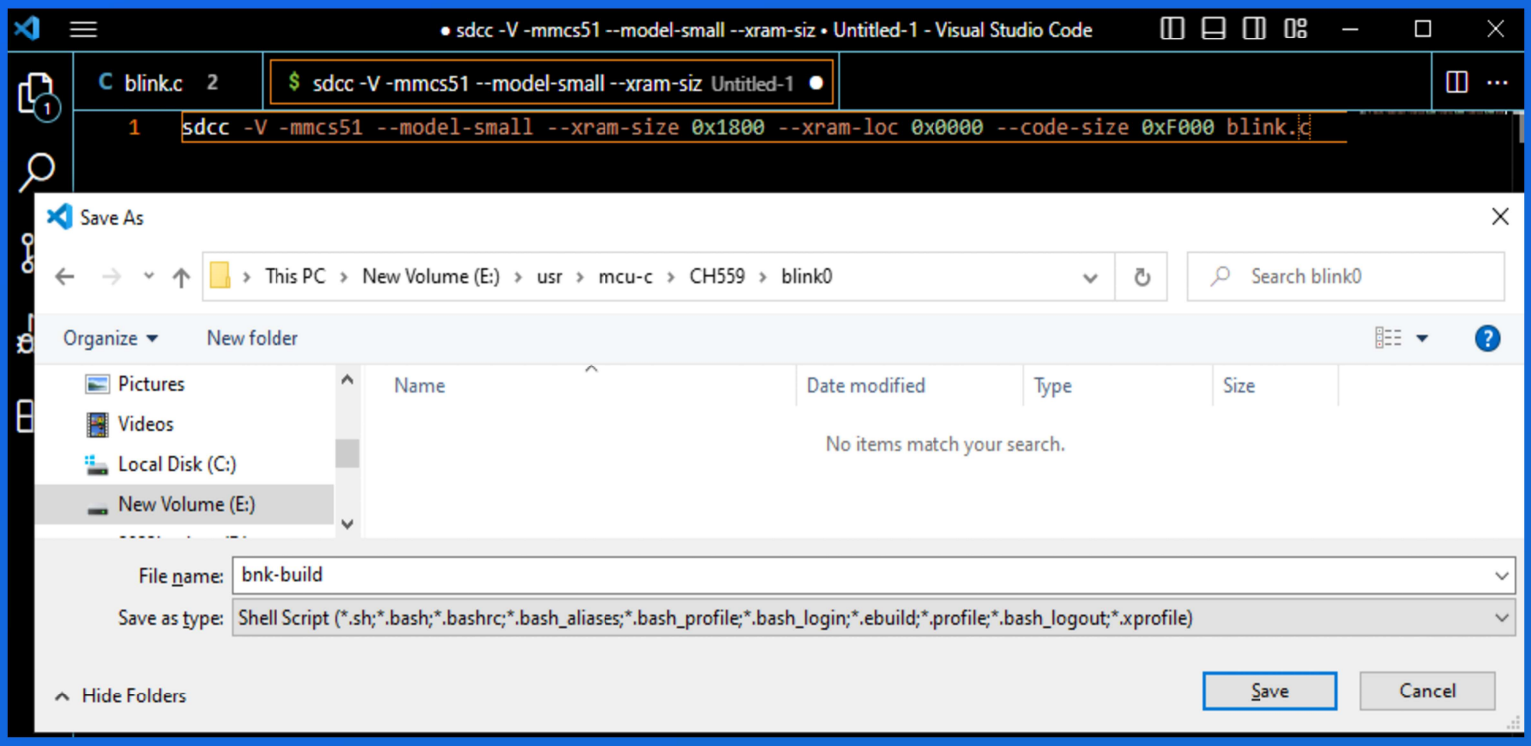
- Since the flash tool is successfully talking to a board, let's go compile a program!
- From desktop icon or from Start menu, open up VS Code.
- In VS Code, paste the code below into a new file and press Ctrl+Shift+S to save as.
- Within the save-as dialog, right click for a context menu and select new | folder. Name it blink0.
- Save the new file blink.c inside the folder blink0.

```
#include <compiler.h>
#include <stdint.h>

SFR(PORT_CFG, 0xC6);          // Port Configuration Register
SFR(P1_DIR,   0xBA);          // P1 port direction control register
SFR(P1,       0x90);          // P1 port input and output register

SBIT(LED6, 0x90, 6);          // accessing pin 1.6


static inline void delay(uint32_t total)
{
  uint32_t i;
  for(i = 0; i < total; i++)   { }
  __asm__("nop");
}


void main()                   // program entry point
{
  PORT_CFG = 0b00101101;
  P1_DIR =   0b11110000;
  P1 = 0x00;

  while(1) {                  // loop forever, no program exit
    if(LED6)   delay(120000UL);
    else       delay(480000UL);
    LED6 = !LED6;             // toggle   0 to 1   1 to 0
  }
}
```

- Also in VS Code, Press Ctrl+N for new file and paste the text below.

```
sdcc -V -mmcs51 --model-small --xram-size 0x1800 --xram-loc 0x0000 --code-size 0xF000 blink.c
```

- Press Ctrl+Shift+S for save as. We'll save in the blink0 folder.
- Call the file "bnk-build" with a file type of "Shell Script."

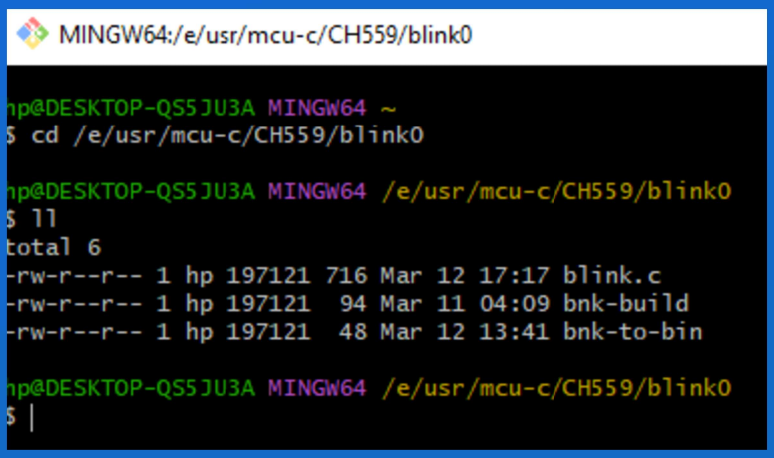


Alright, just one more and we'll be done creating project folder files.

- Also in VS Code, Press Ctrl+N for new file and paste the text below.

```
sdobjcopy -I ihex -O binary blink.ihx blink.bin
```

- Press Ctrl+Shift+S to save this one in the blink0 folder as a file named "bnk-to-bin" with type of "Shell Script."
- Now open the Git Bash shell and navigate to your blink0 folder, similar to my navigation below...

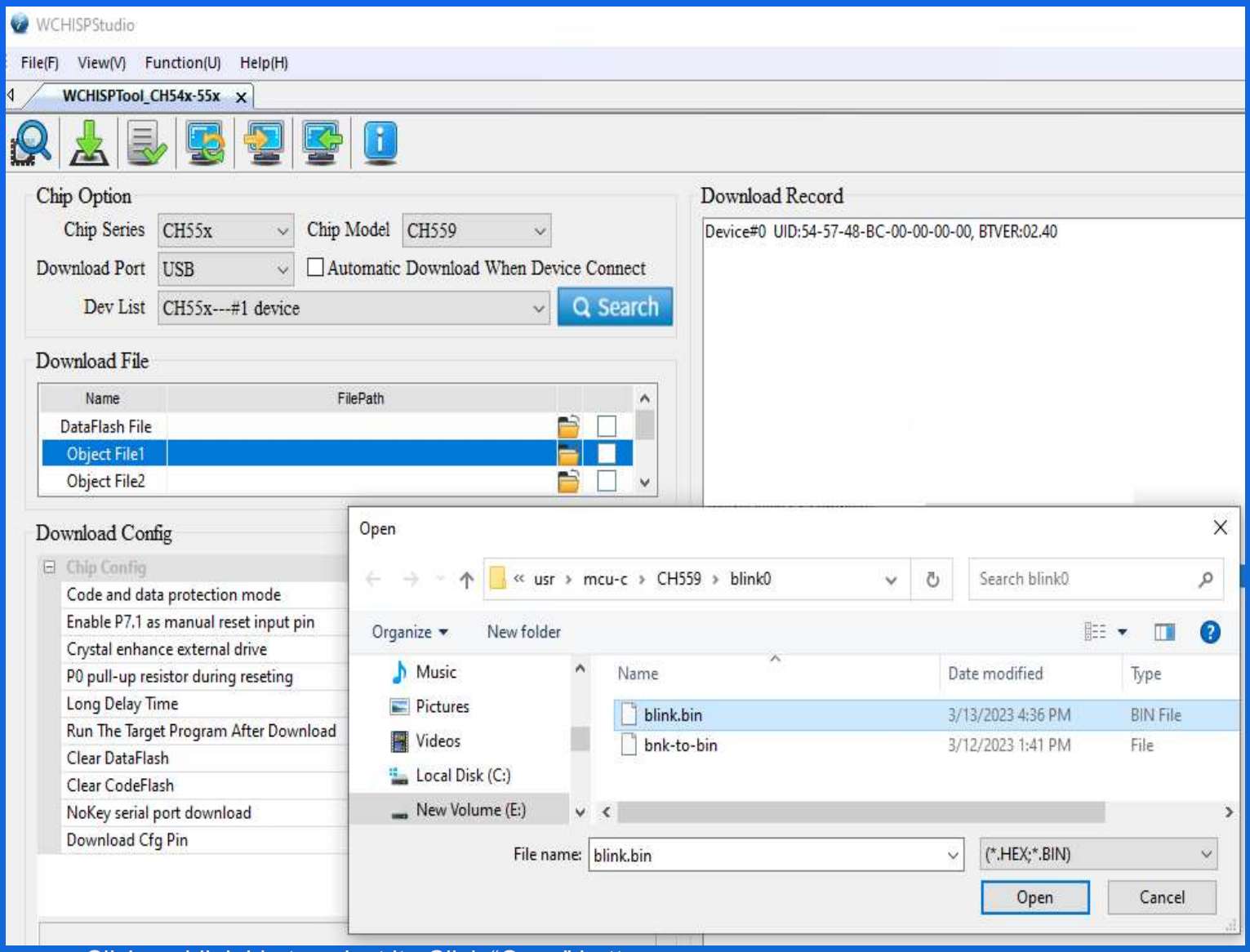


- Provided the SDCC installer set the tool path ok, you should be able compile by typing ./bnk-build

```
hp@DESKTOP-QS5JU3A MINGW64 /e/usr/mcu-c/CH559/blink0
$ ./bnk-build
+ E:\usr\bin\SDCC\bin\sdcpp.exe -nostdinc -Wall -std=c11 --obj-ext=.rel -D__SDCC
_CHAR_UNSIGNED -D__SDCC_MODEL_SMALL -D__SDCC_FLOAT_REENT -D__SDCCCALL=0 -D__SDCC
=4_2_14 -D__SDCC_VERSION_MAJOR=4 -D__SDCC_VERSION_MINOR=2 -D__SDCC_VERSION_PATCH
=14 -DSDCC=4214 -D__SDCC_REVISION=13904 -D__SDCC_mcs51 -D__STDC_NO_COMPLEX__=1 -
D__STDC_NO_THREADS__=1 -D__STDC_NO_ATOMICS__=1 -D__STDC_NO_VLA__=1 -D__STDC_ISO_
10646__=201409L -D__SIZEOF_FLOAT__=4 -D__SIZEOF_DOUBLE__=4 -D__SDCC_BITINT_MAXWI
DTH=64 -isystem "E:\usr\bin\SDCC\bin\..\include\mcs51" -isystem "E:\usr\bin\SDCC
\bin\..\include"  -xc "blink.c"
+ E:\usr\bin\SDCC\bin\sdas8051.exe -plosgffw "blink.rel" "blink".asm
+ E:\usr\bin\SDCC\bin\sdld.exe -nf "blink.lk"
blink.c:19: warning 283: function declarator with no prototype

hp@DESKTOP-QS5JU3A MINGW64 /e/usr/mcu-c/CH559/blink0
$ ll
total 108
-rw-r--r-- 1 hp 197121  7572 Mar 12 18:27 blink.asm
-rw-r--r-- 1 hp 197121   716 Mar 12 17:17 blink.c
-rw-r--r-- 1 hp 197121   684 Mar 12 18:27 blink.ihx
-rw-r--r-- 1 hp 197121   213 Mar 12 18:27 blink.lk
-rw-r--r-- 1 hp 197121 17985 Mar 12 18:27 blink.lst
-rw-r--r-- 1 hp 197121 11585 Mar 12 18:27 blink.map
-rw-r--r-- 1 hp 197121  1192 Mar 12 18:27 blink.mem
-rw-r--r-- 1 hp 197121  2781 Mar 12 18:27 blink.rel
-rw-r--r-- 1 hp 197121 18252 Mar 12 18:27 blink.rst
-rw-r--r-- 1 hp 197121 31632 Mar 12 18:27 blink.sym
-rw-r--r-- 1 hp 197121    94 Mar 11 04:09 bnk-build
-rw-r--r-- 1 hp 197121    48 Mar 12 13:41 bnk-to-bin

hp@DESKTOP-QS5JU3A MINGW64 /e/usr/mcu-c/CH559/blink0
$
```

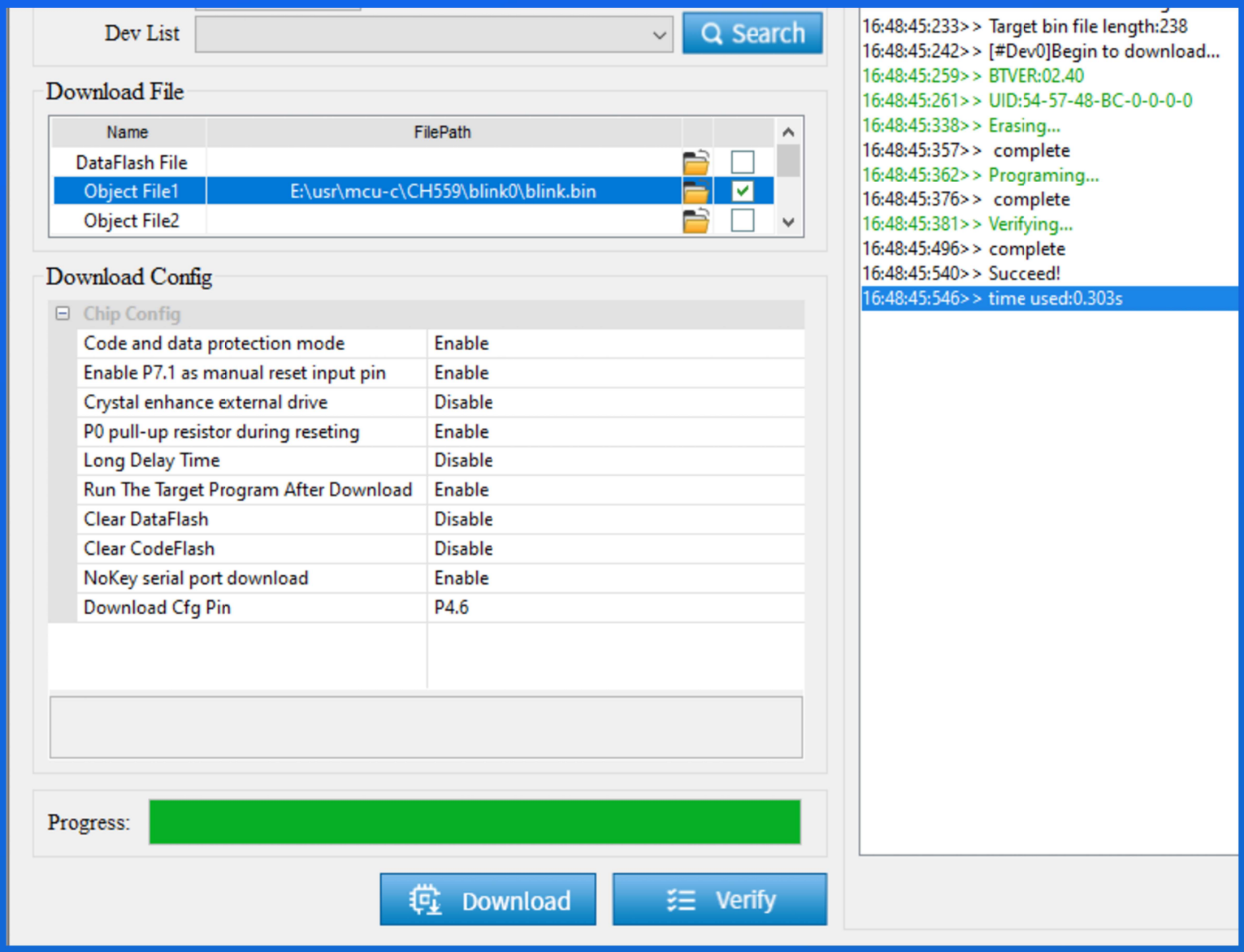
- Now, the output file is blink.ihx
- And it must be converted to blink.bin
- So, next type ./bnk-to-bin and press Enter.
- OK, so we have a binary executable. Let's switch back to the USB flash tool called WCHISPTool.

- Click the folder icon next to “Object File1” to browse files. Navigate to the blink0 folder.



- Click on blink.bin to select it. Click “Open” button.

- Click the “Download” button pictured below to send blink.bin over to the board.



On CH559, the program executes immediately. If you wish to modify the code and flash again, remember to hold the PROG button while applying power to place the board back into program flash mode.

Great job on reaching the last line of this tutorial!