STC 8051 Development Workflow

From SDCC Source to Programmed Microcontroller

Command-line tutorial for Windows
using SDCC and the native `stcisp` tool

Abstract

This note describes a complete, minimal Windows workflow for developing firmware for STC 8051-family microcontrollers. Source is compiled with SDCC; the resulting Intel HEX file is programmed with the native Windows port of `stcisp`. Two short batch files automate the everyday steps.

Contents

1 Overview

The everyday cycle is:

1. Edit a C source file (e.g. `LedBlink.c`).
2. Run `buildhex.bat LedBlink.c` → produces `LedBlink.hex`.
3. Run `program_hex.bat LedBlink.hex` → downloads the image to the MCU.

4. Power-cycle (or reset) the target when the programmer begins sending `0x7F` probes.

Once the tools are on the PATH (or sit in the same folder as the batch files) the whole process is only two commands.

2 Prerequisites

2.1 Software

- **SDCC** (Small Device C Compiler) — the free 8051 compiler.
  Download from https://sdcc.sourceforge.net/.
  After installation, confirm that `sdcc` is on the PATH:

```
sdcc -v
```

- **stcisp.exe** — the native Windows ISP programmer produced from the `stc-isp-win` package. It must be in the same directory as the batch files, or on the PATH.
- A recent Windows command prompt (or PowerShell). No special IDE is required, although WinEdt / any text editor works well for editing the C sources and the optional LaTeX notes.

2.2 Hardware

- STC 8051-family MCU (classic STC89/90 series are the best match for this tool).
- USB-to-TTL serial adapter (or a real COM port) wired as follows:

Adapter / PC	MCU pin
TXD	RXD (P3.0)
RXD	TXD (P3.1)
GND	GND
5 V (optional)	Vcc

- Ability to power-cycle the board or press a reset button while the programmer is probing.

3 Compiling with SDCC — `buildhex.bat`

A typical one-line compile for a small 8051 program is:

```
sdcc -mmcs51 LedBlink.c
```

SDCC produces several files; the one needed for programming is the Intel HEX file (`LedBlink.ihx` or, after a simple rename/conversion, `LedBlink.hex`).

A convenient wrapper is the batch file `buildhex.bat`:

```
@echo off
if "%~1"=="" (
    echo Usage: buildhex.bat source.c
    exit /b 1
)
sdcc -mmcs51 "%~1"
if errorlevel 1 (
    echo Compile failed
    exit /b 1
```

```
)
echo.
echo HEX file ready: %~n1.ihx
```

Usage:

```
buildhex.bat LedBlink.c
```

If your toolchain already converts `.ihx` → `.hex`, either name is accepted by `stcisp`; the formats are identical.

4 Programming — `program_hex.bat` / `stcisp`

4.1 Command-line form

```
stcisp -s COM3 -b 2400 -f LedBlink.hex
```

Option	Meaning	Typical value
`-s`	Serial port	`COM3` (or `1`, `2`, ...)
`-b`	Initial baud rate	`2400` (most reliable)
`-f`	Firmware image	`.hex` / `.ihx` / `.bin`

4.2 Batch-file form

The supplied `program_hex.bat` fixes the port and baud rate so everyday use is simply:

```
program_hex.bat LedBlink.hex
```

Inside the batch file the essential line is:

```
stcisp -s COM3 -b 2400 -f "%~1"
```

(Adjust `COM3` and `2400` if your hardware differs.)

4.3 The power-cycle step

After the programmer opens the port it begins sending a continuous stream of `0x7F` bytes:

```
TX: 7F
TX: 7F
TX: 7F
...
```

At this moment power-cycle the MCU (or press its reset button). The STC boot-loader runs only for a short window after reset; seeing the `0x7F` stream makes it stay in ISP mode.

You should then see messages similar to:

```
Start Programming >>>
target is full gain
target is normal speed(12T)
target's P1.1&P1.0 normal
...
Program OK
Have already encrypt.
```

When "Program OK" appears the flash has been written and the device is ready to run.

5 Recommended everyday workflow

1. Edit the C source.
2. Compile:

```
buildhex.bat MyProgram.c
```

3. Program (leave the serial cable connected):

```
program_hex.bat MyProgram.hex
```

   or, if the batch file still points at a different baud rate,

```
stcisp -s COM3 -b 2400 -f MyProgram.hex
```

4. When the `TX: 7F` stream appears, power-cycle / reset the board.
5. Wait for "Program OK".
6. The new firmware is now running.

For a small program (a few hundred bytes) the whole download after the reset usually finishes in a couple of seconds at 2400 baud.

6 Why 2400 baud?

9600 baud is faster but can be marginal with many USB-to-TTL adapters and longer cables. 2400 baud is the classic STC handshake speed and is far more tolerant of timing jitter and electrical noise. Once contact is established the protocol may negotiate a higher rate, but starting at 2400 maximises the chance of a clean first contact.

If a particular board works reliably at 9600 you may raise the default; otherwise leave it at 2400.

7 Troubleshooting

Symptom	Likely cause / action
Continuous `TX: 7F` only	MCU never entered ISP mode. Power-cycle *while* the probes are running.
"Start Programming" then silence	Partial target-info packet. Retry at 2400 baud; check wiring and ground.
"Cannot open COMx"	Port in use by another program, wrong number, or missing USB-serial driver.
Compile errors from SDCC	Missing `-mmcs51`, syntax error, or SDCC not on PATH.
Program OK but MCU does nothing	Check that the code enables the correct port pins / clock; verify the HEX really contains the intended image.

8 File summary

File	Role
`*.c`	SDCC source
`buildhex.bat`	Compile source → HEX
`*.hex` / `*.ihx`	Intel HEX image
`stcisp.exe`	Native Windows ISP programmer
`program_hex.bat`	One-click download (COM3, 2400 baud)

9 Notes on the programmer itself

`stcisp` is a native Windows port of the classic open-source STC ISP tool. It talks the older STC89/90-style protocol. Newer families (STC15, STC8, ...) often need a different utility such as `stcgal`. For the classic parts used in many hobby and teaching boards the present tool is sufficient and self-contained—no Python runtime or extra DLLs are required.

The source package (`stc-isp-win`) builds with ordinary MinGW-w64 / GCC (for example the GCC 13.2 distribution). The serial layer uses the standard Win32 Communications API.

10 Quick reference card

```
:: Compile
buildhex.bat LedBlink.c

:: Program (then power-cycle when TX: 7F appears)
program_hex.bat LedBlink.hex

:: Or explicitly
stcisp -s COM3 -b 2400 -f LedBlink.hex
```

Once the two batch files are on your desktop or in your project folder, the entire edit–compile–program cycle reduces to those two commands plus a single power-cycle of the target.