Flashing LimiFrog's STM32 over USB

Work in progress

Draft A
02/2016

Introduction

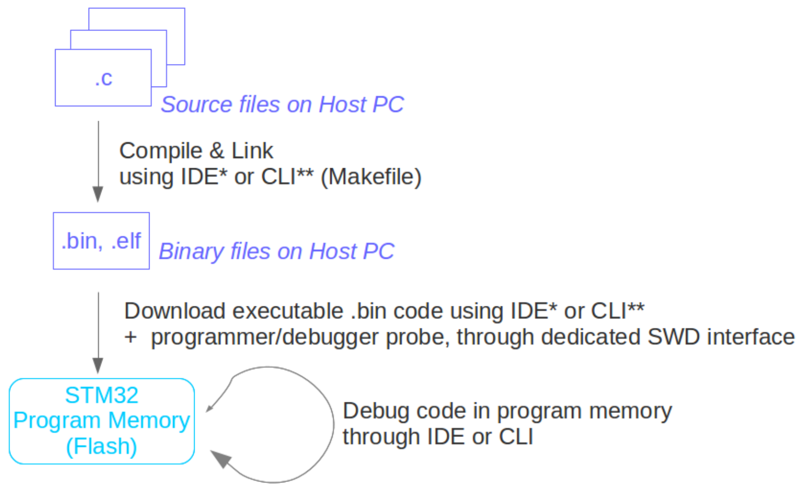
It is possible to download executable code into LimiFrog through a USB cable rather than using the “ST-Link” (which is the programmer/debugger probe intended for the STM32).

Although this saves the need for an external programmer/debugger probe, this comes with some constraints :
- a pure USB link does not provide any debug capability. Once downloaded, the code can only be normally executed: no way to step through the code, insert break points or examine variables, for example.
- a specific software must be used to download the code into the STM32's program memory: this may be ST's utility called DfuSe (Windows only) or an open-source program called dfu-util (Linux, Mac OS or Windows). Refer to Appendix for more information.
- If using DfuSe, then the .bin executable requires a translation from .bin to .dfu using a utility called DFU File Manager
- A specific (user-definable) GPIO must be pulled to Logic 0 at power-up or hardware reset to cause the STM32 to obtain code over USB rather than normally running the program present in Flash memory.
- The source code must start with a specific routine (provided). Flashing a code that does not start with this routine will result in the unability to flash any new code over USB and make it mandatory to use an ST-Link programmer/debugger probe. *[This is due to a hardware limitation of the LimiFrog board]*

Figures 1.a and 1.b compare the option of using the ST-Link programmer/debugger vs downloading code through USB only.

Fig. 1.a :

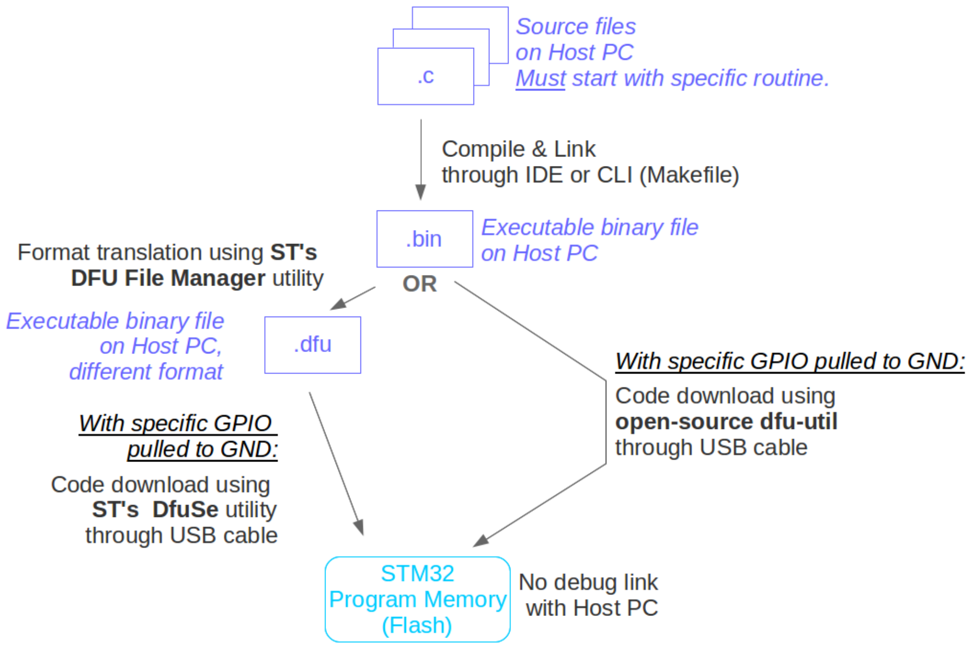
CODE DOWNLOAD AND DEBUG THROUGH SWD INTERFACE


.c
Source files on Host PC
Compile & Link
using IDE* or CLI** (Makefile)
.bin, .elf
Binary files on Host PC
Download executable .bin code using IDE* or CLI**
+ programmer/debugger probe, through dedicated SWD interface
STM32
Program Memory
(Flash)
Debug code in program memory
through IDE or CLI

*: IDE = Integrated Devlopment Environment
**: CLI = Command Line Interface

Fig. 1.b :

CODE DOWNLOAD THROUGH USB


.c
Source files
on Host PC
Must start with specific routine.
Compile & Link
through IDE or CLI (Makefile)
.bin
Executable binary file
on Host PC
OR
Format translation using ST's DFU File Manager utility
Executable binary file
on Host PC,
different format
.dfu
With specific GPIO pulled to GND:
Code download using open-source dfu-util through USB cable
With specific GPIO pulled to GND:
Code download using ST's DfuSe utility through USB cable
STM32
Program Memory
(Flash)
No debug link
with Host PC

Principle of operation

The STM32 includes a “bootloader” code stored in non-volatile memory. When invoked, this bootloader will try to obtain program code from a number of possible sources, including USB, and place it in program memory.
Normally, this code is invoked by pulling pin BOOT0 of the STM32 to Ground during power-up or reset – in which case the STM32 automatically jumps to bootloader execution.
However, because of some interaction between an on-board chip and the STM32 at reset, this does not always work on LimiFrog.
There is a solution : replace this automatic (hardware) mechanism by a short software routine placed at the beginning of any program downloaded into LimiFrog. This routine (provided as part of the LimiFrog software package) simply examines the state of a user-accessible pin of the STM32: if that pin is being pulled to Logic 0 (for example) then the routine branches to the bootloader, else it continues executing code normally.

Steps to follow – In practice

Pre-requisite:
DFU-Util / DfuSe – see Appendix

1) Make sure your application code starts, before anything else, with :
> function `LBF_Init()`, provided in the LimiFrog software package -
or *(if you don't use LBF_Init() :*
> with routine `LBF_DFU_IfNeeded()`, provided in the LimiFrog software package – which is actually the very first routine called by `LBF_Init()` when using the former option,
or *:*
> with a similar routine provided by you – for example, you might want to change the pin that will control execution of the USB bootloader.

CAUTION:
If at any point you download a code that does not start with this kind of routine, then it will be IMPOSSIBLE to download any new code over USB. To flash a new program you will have to use the regular SWD interface, through a STLink dongle. (In this new program you could put back the missing routine to re-enable USB download capability.)

2)
Create a .bin executable of your code by compiling and linking it using your favorite development environment (gcc + Makefiles, IDE...)

2-bis)
If you intend to use ST's DfuSE utility, translate this .bin file to .dfu format using the DFU File Manager utility provided along with DfuSE.
This is because DfuSE does not allow to directly download .bin file, it works with .dfu files

which have some additional information added to the .bin. The expected .dfu file can be generated from the .bin using the “File Manager” utility of DfuSE. See <DFUse User Manual>.

3)
Connect position 10 of the extension connector to Logic 0 : directly to Ground or (safer, in case you make a connection mistake) through a resistor. Then power-up / reset LimiFrog. After reset, that position can take any value.

At this point, the STM32 enters the bootload mode rather than executing normally the code present in program memory at that time.
If Position 10 of the extension port is not grounded at power-up/reset, then the code already present in program memory executes normally.

4)
Connect a USB cable between LimiFrog and the PC.

5)

<u>5.a: If you use dfu-util -</u>

* Optional : check dfu-util can see your device in DFU mode
For example by typing in a console window : *`sudo dfu-util -l`*
(sudo can be omitted depending on your permissions settings).
The answer should look like this :
.....

* Perform the actual code download, by typing :

```
sudo dfu-util -a 0 -s 0x08000000:leave -D /path/to/file.bin
```

> 0x0800000 is the start address in program memory at which the code will be placed
> Option *:leave* tells *dfu-util* to exit after completing the code download: the STM32 reboots (and will execute the new program <u>provided you have removed the connection between extension port position 10 and ground</u> – else it re-enters bootloader mode).
> <*`path/to/file.bin`*> represents the path to the .bin executable you want to download

See dfu-util manual for options and variants or to adapt to your specific case.

<u>5.b: If you use DfuSE -</u>

When you connect the USB cable, the DfuSe application should report the presence of a new DFU device.

Download into LimiFrog the .dfu file you have previsouly generated, as follows:
<To Do>

APPENDIX : dfu_util and DfuSe utilities
<To Do>

References
<To Do>