

AN00135: Interfacing FlashRunner with STMicroelectronics ST72F Devices

by Pietro Poletto (pietro.poletto@softecmicro.com)

FlashRunner is a Universal In-System Programmer which uses the principles of In-Circuit Communication (ICC) and In-Circuit Programming (ICP) to program STMicroelectronics ST7F microcontrollers. This application note describes how to properly set up and use FlashRunner to program ST7 Flash devices.

This Application Note assumes that you are familiar with FlashRunner and with the main features of the ST7F family. Full documentation on these topics is available in device-specific datasheets and in the FlashRunner user's manual.

1. Introduction

Two Flash technologies are available in the ST7F family with different programming methods.

- **XFlash (Extended Flash):** Extended Flash memory is based on EEPROM technology. XFlash provides extended features such as byte-by-byte re-programming (by means of byte erasing) and data EEPROM capability. XFlash devices are available between 1KB and 16KB. Anything present in this memory zone is automatically erased when you program or reprogram this type of memory. Because of this, Erase and Blank-Check operations are not necessary with XFlash memory.



Copyright © 2007 SofTec Microsystems®
Revision 1.1 – December 2007

SofTec Microsystems is a STMicroelectronics Design Consultant.



DC01483

SofTec Microsystems

E-mail (general information): info@softecmicro.com

E-mail (technical support): support@softecmicro.com

Web: <http://www.softecmicro.com>

Important

SofTec Microsystems reserves the right to make improvements to its products, their documentation and software routines, without notice. Information in this manual is intended to be accurate and reliable. However, SofTec Microsystems assumes no responsibility for its use; nor for any infringements of rights of third parties which may result from its use.

SOFTEC MICROSYSTEMS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

Trademarks

SofTec Microsystems™ and the SofTec Microsystems logo are trademarks of SofTec Microsystems.

Other products and company names listed are trademarks or trade names of their respective companies.

- **HDFlash (High-Density Flash):** High-density Flash memory is based on Flash technology. The high-density of the HDFlash cell is used for devices from 4KB up to 60KB. HDFlash is programmed byte per byte but erased by sector. You cannot erase single bytes—only sectors. Before programming byte(s) or sector(s) you must erase the appropriate sector(s). You can perform a Blank Check operation to verify that the correct sectors are erased.

The following table shows a classification of ST7F devices based on memory type.

| Flash Technology | ST7 Devices | Prog. Methods |
|------------------|--|---------------|
| XFlash | ST7Flitex, ST7FDALI, ST7FLx, ST72F62x, ST72F34x | ICP, IAP |
| HDFlash | ST72F32x/F521, ST72F561, ST72F361, ST72F60x, ST72F61x, ST72F62x, ST72F63x, ST72F63Bx, ST72F65, ST72FHUB, ST72FSCR, ST72FMC | ICP, IAP |

Thanks to its in-system programming capabilities, FlashRunner allows you to program or update the content of the Flash memory when the chip is already plugged on the application board.

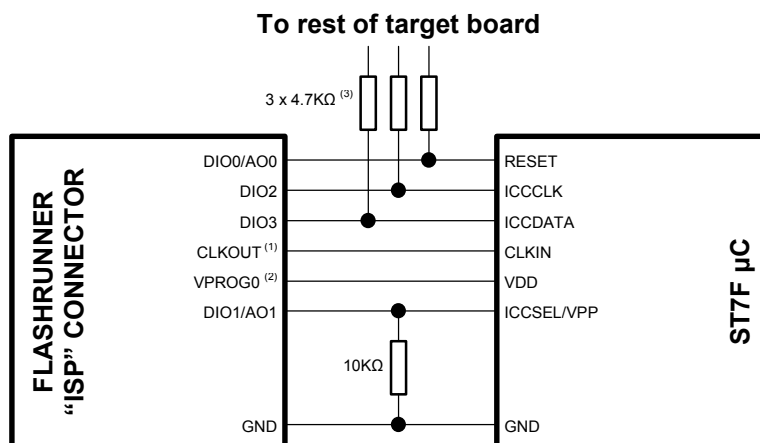
In order to perform in-system programming, the ST7 microcontroller must be switched to a special functional mode called ICC (In-Circuit Communication) mode by FlashRunner. When using FlashRunner to perform in-system programming, you need to implement the ICP hardware interface on your application board.

2. Hardware Configuration

The microcontroller lines needed to program an ST7 device in ICC mode are the following:

- **ICCDATA:** In-Circuit Communication input/output serial data pin.
- **ICCCLK:** In-Circuit Communication serial clock input/output pin.
- **ICCSEL/VPP:** ICC Entry Mode/Programming voltage. Please note that, on some devices (e.g. ST7FLITE0), the ICCSEL/VPP line is not available.
- **CLOCK:** Used to provide a device input clock signal to use when no other clock sources are available. FlashRunner provides a 6.25, 12.5 or 25MHz signal on this pin.
- **RESET:** Device reset (input/output).
- **VSS:** Device power supply ground.
- **VDD:** Device power supply voltage.

These lines must be connected to the FlashRunner's "ISP" connector according to the following diagram:



Notes

- (1) Connect this line if you want FlashRunner to automatically power the target device
- (2) This connection is not needed if you use the **TCSETPAR ICPMODE ENABLED** command and the clock source specified by the Option Bytes is provided by the target board
- (3) If the RESET, ICCCLK or ICCDATA lines are used for other purposes in the application, 4.7KΩ series resistors (according to STMicroelectronics specifications) should be implemented to avoid a conflict in case the rest of the target board forces the signal level. If these lines are used as outputs, these resistors are not necessary.

3. ST7-Specific TPSETPAR Programming Commands

Overview

TPSETPAR commands set device-specific and programming algorithm-specific parameters. These commands must be sent after the **TCSETDEV** command and before a **TPSTART / TPEND** command block.

All of the FlashRunner programming capabilities rely on the ICC (In-Circuit Communication) mode of the target device. In order to enter this special mode (which establishes a communication channel between the target device and FlashRunner through the ICCDATA and ICCCLK lines) and configure it properly, all of the following parameters must be correctly specified through the relative **TPSETPAR** commands (although the order with which these parameters are set is not important):

- ICC mode;
- V_{DD} ;
- Reset up time;
- Reset down time;
- Power up time (optional);
- Power down time (optional);
- CLKOUT frequency.

TPSETPAR ICPMODE

Command syntax:

```
TCSETPAR ICPMODE ENABLED | DISABLED
```

Command options:

ENABLED: Enters ICC mode taking into account the Option Bytes.

DISABLED: Enters ICC mode ignoring the Option Bytes.

Description:

Specifies how FlashRunner will enter the ICC mode. You can choose to use the Option Bytes programmed into the device or to bypass them and use the Option Bytes' default value instead (the value that would be used if the device were blank). In the former case, you must provide the required clock source specified in the Option Bytes value programmed into the device—otherwise the device will not work. In the latter case, the device is forced to use its default Option Bytes value. This means, among other things, that a “safe” clock source is used (typically an external clock must be provided). This option is useful when the Option Bytes value programmed into the device specify incorrect start-up parameters; in this case, you can bypass them and start-up from a safe condition.

TPSETPAR VDD

Command syntax:

```
TCSETPAR VDD <voltage mV>
```

Parameters:

voltage mV: Target device supply voltage, expressed in millivolts.

Description:

This command is used to properly generate the voltage level of the ICCDATA, ICCCLK, OSC_CLK and RESET signals. Additionally, the specified voltage is routed to the VPROG0 line of the FlashRunner “ISP” connector, which can be used as a supply voltage for the target board.

TPSETPAR RSTUP

Command syntax:

```
TCSETPAR RSTUP <time μs>
```

Parameters:

time μs: Reset rising time, expressed in microseconds.

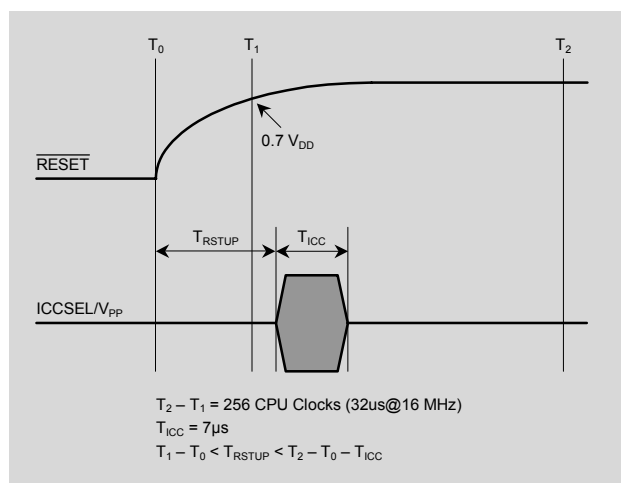
Description:

This command is necessary because, to enter the ICC mode, FlashRunner must properly drive the RESET pin (see ICC Protocol reference manual for details).

The Reset rising time (RSTUP) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the high logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the high logic level. If RSTUP isn't long enough, FlashRunner could not be able to enter the ICC mode.

RSTUP is used to determine the moment after which a series of ICCSEL/VPP pulses will start. This series of pulses must conclude within 256 CPU cycles from the moment when the Reset signal has reached the high logic level—otherwise the device won't enter the ICC mode. For this reason you cannot enter too large a value for RSTUP.

The following figure shows you how to properly choose RSTUP.



TPSETPAR RSTDOWN

Command syntax:

TCSETPAR RSTDOWN <time µs>

Parameters:

time µs: Reset falling time, expressed in microseconds.

Description:

The Reset falling time (RSTDOWN) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the low logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the low logic level.

TCSETPAR PWUP

Command syntax:

```
TCSETPAR PWUP <time ms>
```

Parameters:

time ms: Power rising time, expressed in milliseconds.

Description:

This command is necessary because, to enter the programming mode, FlashRunner must properly drive the V_{DD} line during the power-on reset.

The V_{DD} rising time (PWUP) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the high logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the high logic level. If PWUP is not long enough, FlashRunner could not be able to enter the programming mode.

TCSETPAR PWDOWN

Command syntax:

```
TCSETPAR PWDOWN <time ms>
```

Parameters:

time ms: Power falling time, expressed in milliseconds.

Description:

The V_{DD} falling time (PWDOWN) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the low logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the low logic level.

TPSETPAR CLKOUT

Command syntax:

```
CSETPAR CLKOUT 25000000 | 12500000 | 6250000 | 0
```

Command options:

Frequency of a clock signal to be generated at the CLOCKOUT pin of the FlashRunner "ISP" connector, expressed in Hertz.

Description:

Generates an auxiliary clock signal at the CLOCKOUT pin of the FlashRunner "ISP" connector. This signal can be used to enter the target device's ICC mode (ignoring the Option Bytes) when the target device

does not have an external clock. Furthermore, this signal can be used to speed up programming (when you want to use a clock faster than that provided by your target board).

Make sure that the clock frequency you select isn't greater than the maximum allowed frequency for your device. If the target device has an internal frequency divider, the actual device's frequency will be a fraction of the CLKOUT frequency.

If you specify 0 as the CLKOUT frequency, no clock signal is generated.

4. ST7-Specific TPCMD Programming Commands

Overview

TPCMD commands perform a programming operation (i.e. mass erase, program, verify, etc.) These command must be sent within a **TPSTART** / **TPEND** command block.

The ST7-specific target programming commands are the following:

- **TPCMD UNPROTECT;**
- **TPCMD BLANKCHECK;**
- **TPCMD MASSERASE;**
- **TPCMD PROGRAM;**
- **TPCMD PROGRAM OPTIONS;**
- **TPCMD RESTORE_RC_CALIBRATION;**
- **TPCMD TRIM;**
- **TPCMD READ OPTIONS;**
- **TPCMD VERIFY;**
- **TPCMD VERIFY OPTIONS;**
- **TPCMD RUN.**

TPCMD UNPROTECT

Command syntax:

```
TCSETPAR UNPROTECT
```

Parameters:

None.

Description:

Unprotects a read protected device. Sets the bit of the Option Byte(s) representing the read protection. Unprotecting a device causes its contents to be lost.

TPCMD BLANKCHECK

Command syntax:

```
TPCMD BLANKCHECK F|E <tgt start addr> <len>
```

Command options and parameters:

F|E: Memory type to be blankchecked. **F** stands for Flash program memory and **E** stands for EEPROM data memory.

tgt start addr: Memory location from which the blankcheck operation will start.

len: Number of bytes to be blankchecked.

Description:

Blankchecks Flash or EEPROM memory. Blankchecks **len** bytes starting from the address specified by **tgt start address**.

ST7 XFlash devices do not need to be blankchecked, because they are automatically erased during programming.

TPCMD MASSERASE

Command syntax:

```
TPCMD MASSERASE F|E
```

Command options:

F|E: Memory type to be mass erased. **F** stands for Flash program memory and **E** stands for EEPROM data memory.

Description:

Mass erases Flash or EEPROM memory.

Please note that, on ST7 HDFlash devices, the TPCMD MASSERASE command erases the memory only if the device is not blank. ST7 XFlash devices do not need to be mass erased, because they are automatically mass erased during programming.

Please also note that, after a device is mass erased, RC calibration values are lost. Thus, if your application uses the device's internal clock, it is suggested that, after programming, you restore the RC calibration values.

TPCMD PROGRAM

Command syntax:

```
TPCMD PROGRAM F|E <src offset> <tgt start addr> <len>
```

Command options and parameters:

| | |
|------------------------|--|
| F E: | Memory type to be programmed. F stands for Flash program memory and E stands for EEPROM data memory. |
| src offset: | Offset from the beginning of the .FRB source file. |
| tgt start addr: | Memory location from which the program operation will start. |
| len: | Number of bytes to be programmed. |

Description:

Programs **len** bytes in the Flash or EEPROM memory starting from the **tgt start addr** address.

TPCMD PROGRAM OPTIONS

Command syntax:

```
TPCMD PROGRAM OPTIONS <B0> [<B1>]
```

Command parameters:

| | |
|----------------|---|
| B0, B1: | Value(s) of the Option Byte(s) to be programmed. B1 is optional because not all devices have two Option Bytes. |
|----------------|---|

Description:

Programs the Option Byte(s) with the specified values.

Particular attention must be paid in correctly setting each bit of the Option Bytes. Improper settings may cause the target microcontroller not to work correctly (or not to work at all). On some devices, if the Option Bytes are programmed with wrong values, you won't be able to access the device.

In this case you can try to re-program the Option Bytes in the following way (this won't work for all devices, though):

1. Start a new programming block using **TCSETPAR ICPMODE DISABLED**. This will enter the ICC mode ignoring the Option Bytes values.
2. The target device will now use a "default" clock source (typically an external clock). If your target board does not provide such a clock source, you can use the CLKOUT signal of the FlashRunner "ISP" connector.
3. You can now re-program the Option Byte(s) and/or the device memory.

TPCMD READ OPTIONS

Command syntax:

TPCMD READ OPTIONS

Command parameters:

None.

Description:

Reads Option Byte(s).

TPCMD VERIFY

Command syntax:

TPCMD VERIFY F|E R <src offset> <tgt start addr> <len>

Command options and parameters:

F|E: Memory type to be verified. **F** stands for Flash program memory and **E** stands for EEPROM data memory.

R: Specifies Read-Out method. Mandatory option.

src offset: Offset from the beginning of the .FRB source file.

tgt start addr: Memory location from which the verify operation will start.

len: Number of bytes to be verified.

Description:

Verifies that the contents of the target device memory correspond to that of the source image file.

TPCMD VERIFY OPTIONS

Command syntax:

TPCMD VERIFY OPTIONS <B0> [<B1>]

Command parameters:

B0, B1: Option Byte(s) to be verified. **B1** is optional because not all devices have two Option Bytes.

Description:

Verifies that one or both Options Byte(s) correspond to the specified value(s).

TPCMD RESTORE_RC_CALIBRATION

Command syntax:

```
TPCMD RESTORE_RC_CALIBRATION
```

Command parameters:

None.

Description:

Restores the value of the RC calibration bytes to their default factory values.

Note: *this command will be successful only if the `TCSETPAR ICPMODE DISABLED` command (ignore Option Bytes) has been sent.*

Note: *after the device is mass erased, the RC calibration values are lost. Thus, if your application uses the device's internal clock, it is suggested that, after programming, you restore the RC calibration values.*

TPCMD TRIM

Command syntax:

```
TPCMD TRIM <frequency Hz> <tgt start addr>  
          <tolerance_pct>
```

Command parameters:

frequency Hz: Value of the desired frequency, expressed in Hertz.

tgt start addr: Address of the memory location where the trimming value is to be programmed.

tolerance_pct: Precision of trimming value calculation, expressed as a percentage (1 to 100).

Description:

This command trims (calibrates) the device's internal oscillator, if present. It calculates the trimming value for the frequency specified by the **frequency Hz** parameter (to a precision specified by the **tolerance_pct** parameter), and programs it at the **addr** address.

Note: *this command will be successful only if the `TCSETPAR ICPMODE ENABLED` command (use Option Bytes) has been sent.*

TPCMD RUN

Command syntax:

TPCMD RUN

Command parameters:

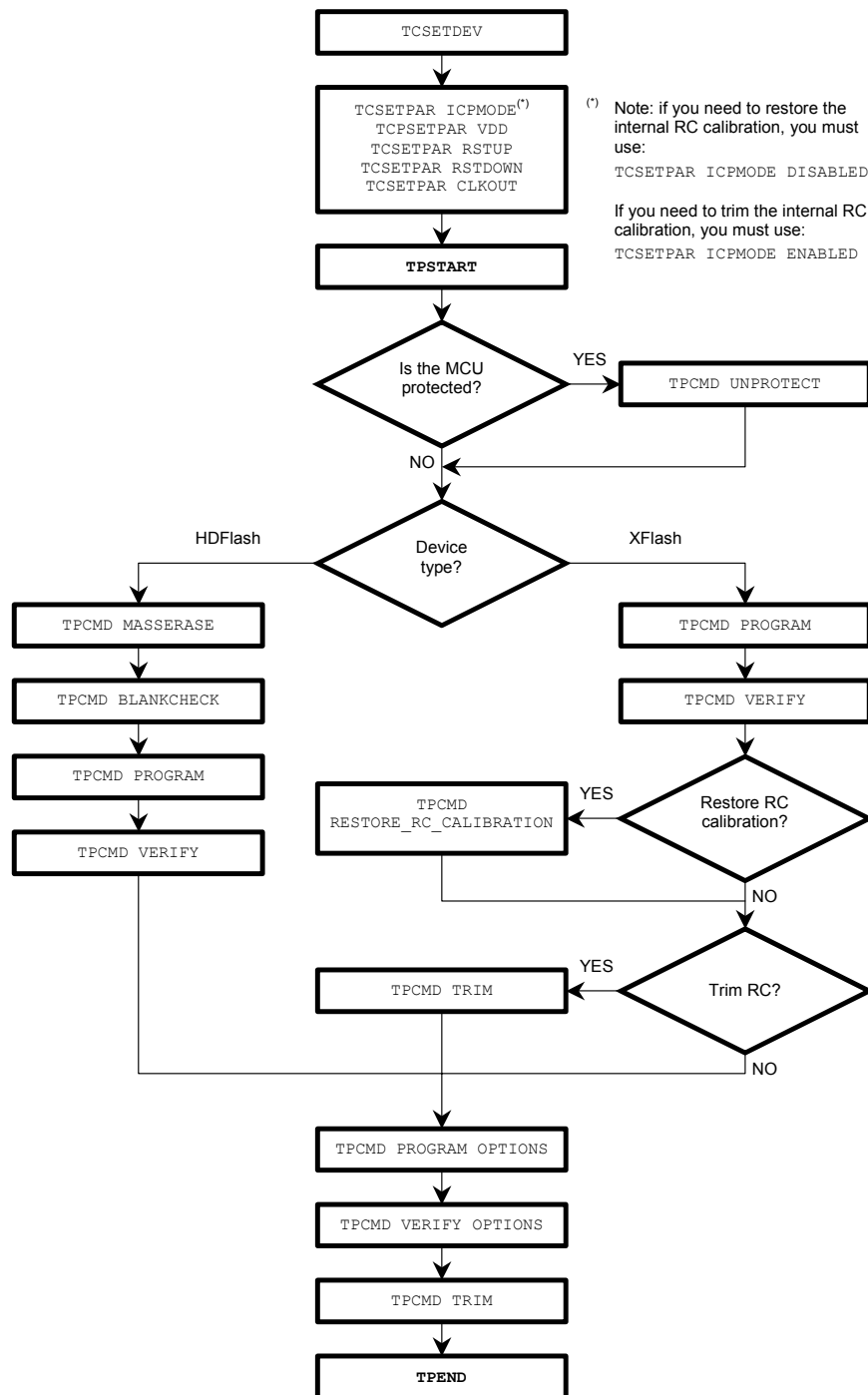
None.

Description:

Runs the target application.

5. Typical Programming Flow

The following flow chart illustrates typical steps to help you write your own script file.



6. Script Examples

Script Example for an HDFlash Device

The example provided below will help you understand how the commands discussed above should be used for a typical HDFlash device, in this case the ST72F324.

```

;
; FLASHRUNNER SCRIPT EXAMPLE FOR STMICROELECTRONICS ST72F324BJ2
;
; Use this example as a starting point for your specific programming needs
;
; -----
;
; Hardware connections
;
; DIO0/AO0 (RESET)
; DIO1/AO1 (ICCSEL/Vpp)
; DIO2      (ICCCCLK)
; DIO3      (ICCDATA)
; CLKOUT    (CLOCK - optional)
;

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON_FAIL

; Sets device
TCSETDEV STMICROELECTRONICS ST72F324BJ2 ST7F

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; Clock oscillator frequency, Hz
; Possible frequencies are: 25000000, 12500000, 6250000, 0 (DISABLED)
TCSETPAR CLKOUT 12500000

; Enters ICC mode ignoring the Option Bytes configuration
TCSETPAR ICPMODE DISABLED

; Reset up and down times
TCSETPAR RSTUP 100 ; us
TCSETPAR RSTDOWN 100 ; us

; Starts programming block
TPSTART

; Image file to be programmed (must be placed in the \BINARIES directory of the SD card)
TPSETSRC FILE TEST.FRB

; Chip unprotection (use this command if the device is protected)
TPCMD UNPROTECT

; Mass erases Flash memory
TPCMD MASSERASE F

; Blank checks Flash memory (change address and length as needed)
TPCMD BLANKCHECK F $E000 8192

; Programs Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $E000 $E000 8192

; Verifies Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $E000 $E000 8192

; Programs and verifies Option Bytes
; The values shown are the factory default settings for this device
TPCMD PROGRAM OPTIONS $E7 $EF
TPCMD VERIFY OPTIONS $E7 $EF

; Ends programming block
TPEND

```

Script Example for an XFlash Device

The example provided below will help you understand how the commands discussed above should be used for a typical XFlash device, in this case the ST7FLITE09.

```

;
; FLASHRUNNER SCRIPT EXAMPLE FOR STMICROELECTRONICS ST7FLITE09
;
; Use this example as a starting point for your specific programming needs
;
; -----
;
; Hardware connections
;
; DIO0/AO0 (RESET)
; DIO1/AO1 (ICCSSEL/Vpp)
; DIO2      (ICCCCLK)
; DIO3      (ICCDATA)
; CLKOUT    (CLOCK - optional)
;

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV STMICROELECTRONICS ST7FLITE09 ST7F

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; Clock oscillator frequency, Hz
; Possible frequencies are: 25000000, 12500000, 6250000, 0 (DISABLED)
TCSETPAR CLKOUT 12500000

; Enters ICC mode ignoring the Option Bytes configuration
TCSETPAR ICPMODE DISABLED

; Reset up and down times
TCSETPAR RSTUP 100 ; us
TCSETPAR RSTDOWN 100 ; us

; Starts programming block
TPSTART

; Image file to be programmed (must be placed in the \BINARIES directory of the SD card)
TPSETSRC FILE TEST.FRB

; Chip unprotection (use this command if the device is protected)
TPCMD UNPROTECT

; Programs Flash memory (change addresses and length as needed)
; Since the device features a XFlash memory, neither a mass erase nor a blankcheck operation
; is necessary.
; The device contains an internal RC oscillator.
; Predefined calibration values are stored in Flash memory for 3V and 5V VDD supply voltages at 25°C.
; $FFDE = RCCR0 (VDD=5V), $FFDF = RCCR1 (VDD=3V).
; If RC calibration is not used, these locations can be used as normal memory.
TPCMD PROGRAM F $FA00 $FA00 1502
TPCMD PROGRAM F $FFE0 $FFE0 32

; Verifies Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $FA00 $FA00 1502
TPCMD VERIFY F R $FFE0 $FFE0 32

; Programs EEPROM memory (change addresses and length as needed)
TPCMD PROGRAM E $1000 $1000 128

; Verifies EEPROM memory, read-out method (change addresses and length as needed)
TPCMD VERIFY E R $1000 $1000 128

; Programs and verifies Option Bytes
; The values shown are the factory default settings for this device
TPCMD PROGRAM OPTIONS $FC $EF
TPCMD VERIFY OPTIONS $FC $EF

; Ends programming block
TPEND

```

7. Programming Times

The following table shows programming times for selected STMicroelectronics ST72F family devices.

| Device | Mem. Size | Clock Frequency | Operations | Time |
|--------------------|--------------|--------------------------------|------------------|---------|
| ST72F324J6 | 32KB HDFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 4.68 s |
| ST72F32AJ2 | 8KB HDFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 1.53 s |
| ST72F344S4 | 16KB XFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 4.70 s |
| ST72F561K9 | 60KB HDFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 8.10 s |
| ST72F621L4 | 16KB HDFlash | $f_{\text{CPU}} = 3\text{MHz}$ | Program + Verify | 4.60 s |
| ST7FLITE09 | 1.5KB XFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 0.80 s |
| ST7FLITE29 | 8KB XFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 2.50 s |
| ST7FLITEUS5 | 1KB XFlash | $f_{\text{CPU}} = 8\text{MHz}$ | Program + Verify | 0.66 s |
| ST7FMC2S6 | 32KB HDFlash | Internal Clock | Program + Verify | 10.30 s |

Programming times depend on Programming Algorithm version, target board connections, target microcontroller mask, and other conditions. Programming times for your actual system may therefore be different than the ones listed here. SofTec Microsystems reserves the right to modify Programming Algorithms at any time.

8. References

FlashRunner User's Manual
 ST7 Family ICC Protocol Reference Manual
 ST7 Family Flash Programming Reference Manual
 Microcontroller-specific datasheets