AN00139: Interfacing FlashRunner with NEC 78K Devices

by Daniele Genero (daniele.genero@softecmicro.com)

FlashRunner is a Universal In-System Programmer which uses the principles of In-Circuit Serial Programming to program NEC 78K microcontrollers. This application note describes how to properly set up and use FlashRunner to program 78K0, 78K0R and 78K0S Flash devices.

This Application Note assumes that you are familiar with FlashRunner and with the main features of the 78K family. Full documentation about these topics is available in the FlashRunner user's manual and in device-specific datasheets.

1. Introduction

NEC 78K devices can be divided into two subfamilies, based on the Flash technology:

- Dual Voltage Flash devices: these devices need a dedicated Flash programming voltage (usually called VPP).
- Single Voltage Flash devices: these devices require no dedicated programming voltage.

In-system programming of 78K microcontrollers is performed by entering the device's programming mode, which allows the programming of the MCU memory, through a synchronous or asynchronous serial protocol (depending on the specific device). The following table shows the programming protocols available for each 78K device.

Core	Family	Device	Single/ Dual Voltage	Flash Technology	Protocols Available	Baudrate [BAUDRATE] (bps)	Serial Clock [SCLK] (Hz)	Oscillator Frequency [FOSC] (Hz)	Notes
78K0	Lx2	UPD78F03xx	S	MF2	UART2, CSI	115200	2400 to 2500000	2000000 to 20000000	Oscillator frequency is fixed to 8000000 in CSI communication mode (the internal oscillator is used)
	Kx2	UPD78F05xx							
	Fx2	UPD78F088x UPD78F089x							
	Kx1+	UPD78F01xxH	S	CZ6HSF	UART2, CSI, CSIHS	153600, 76800, 38400, 31250, 19200 or 9600	2400 to 2500000	2000000 to 16000000	
	Fx1+	UPD78F087x			UART2, CSIHS			4000000 to 16000000	
	Kx1	UPD78F01xx	D	CZ6	UART2, CSI or CSIHS, depending on target device	76800, 38400, 31250, 19200 or 9600	Depending on target device	Depending on target device	
	-	UPD78F00xx UPD78F0233 UPD78F0338 UPD78F0354 UPD78F07xx UPD78F0818 UPD78F0828 UPD78F0828B UPD78F0852 UPD78F09xx							
78K0S	Kx1+	UPD78F92xx UPD78F95xx	S	CZ6HSF	UART1	Depending on oscillator frequency	-	Typically 8M±1%, 6M±1%, 9M±1% or 10M±1%	Verify operation not supported
	-	UPD78F90xx UPD78F91xx UPD78F93xx UPD78F94xx UPD78F98xx	D	CZ6	UART2 or CSI, depending on target device	153600, 76800, 38400, 31250, 19200 or 9600	Depending on target device	Depending on target device	
78K0R	Kx3	UPD78F11xx	S	MF2	UART1	115200, 250000, 500000, 1M or 2M	-	8000000 (internal)	

UART1: 1-wire asynchronous communication
UART2: 2-wire asynchronous communication
CSI: synchronous communication
CSIHS: synchronous communication with handshaking

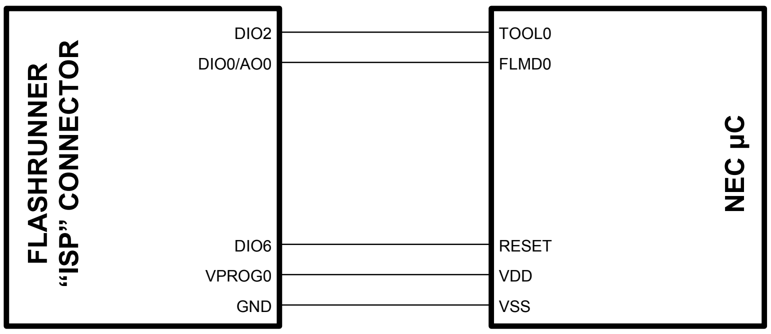
To use FlashRunner to perform in-system programming, you need to implement the appropriate in-circuit programming hardware interface on your application board.

2. Hardware Configuration

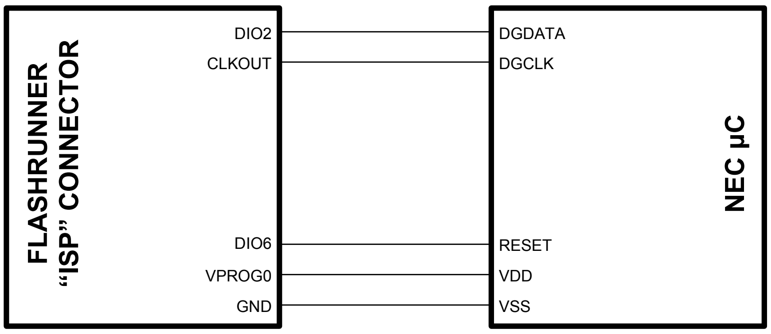
Depending on the device and the communication protocol (see table above), one of the four connection diagrams below need to be implemented.

Note: *if some microcontroller lines are shared with other peripherals/devices, please refer to the microcontroller-specific datasheet for connection guidelines.*

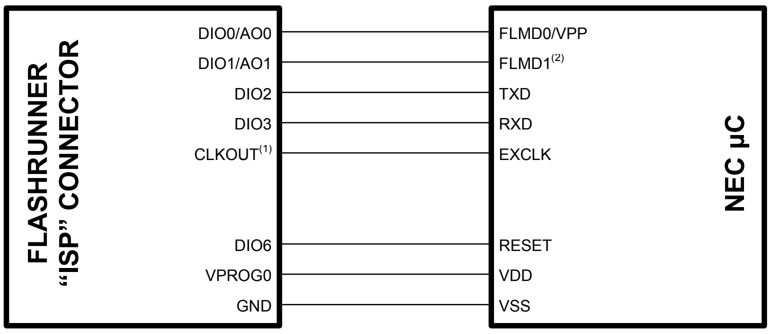
78K0R Connections



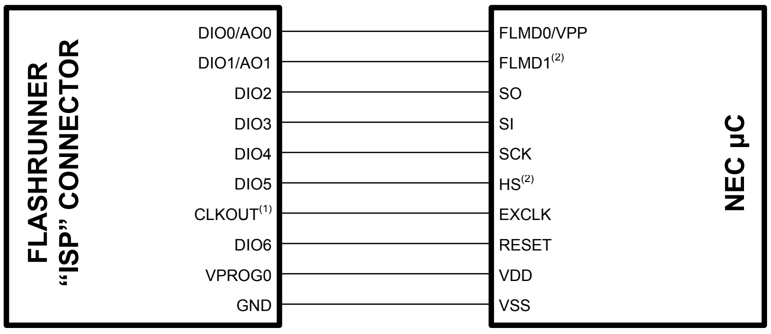
78K0S (Kx1+ Only) Connections



78K0 and 78K0S Connections (UART Communication)



78K0 and 78K0S Connections (CSI Communication)



Notes

(1) Connect this line only if you want the target device to be clocked by FlashRunner.

(2) Connect this line only if available.

3. Specific TCSETPAR Programming Commands

Overview

`TCSETPAR` 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.

In order to enter the programming mode (which establishes a communication channel between the target device and FlashRunner) and configure it properly, the following parameters must be correctly specified through the relative **`TCSETPAR`** commands (although the order with which these parameters are set is not important):

- Communication mode (UART, CSI or CSIHS);
- Baudrate (for UART mode);
- Serial clock (for CSI and CSIHS modes);
- Oscillator frequency;
- V_{DD};
- Auxiliary V_{DD} (if necessary);
- Power up time;
- Power down time;
- Reset up time;
- Reset down time;
- Reset driving mode (push pull or open drain);
- FlashRunner clock out signal.

TCSETPAR CMODE

Command syntax:

```
TCSETPAR CMODE CSI|CSIHS|UART
```

Command options:

`CSI`:	Uses the synchronous communication mode.
`CSIHS`:	Uses the synchronous communication mode with handshaking.
`UART`:	Uses the 1-wire or 2-wire asynchronous communication mode.

Description:

Sets the communication protocol. Please refer to the table on page 2 for the communication protocols supported by your target device.

TCSETPAR BAUDRATE

Command syntax:

```
TCSETPAR BAUDRATE <baudrate>
```

Parameters:

`<baudrate>`:	UART communication speed, in bits per second.

Description:

This command is used to set the communication speed for the UART communication protocol. Please refer to table on page 2 for a list of allowed baudrates for your specific target devices.

Note: when programming 78K0 Lx2, Kx2 and Fx2 devices, a value of 115200 is always used as baudrate and the `TCSETPAR BAUDRATE` command is ignored.

Note: when programming 78K0S Kx1+ devices, the baudrate is automatically calculated by FlashRunner based on the `FOSC` parameter. The `TCSETPAR BAUDRATE` command is ignored.

TCSETPAR SCLK

Command syntax:

```
TCSETPAR SCLK <frequency Hz>
```

Parameters:

`frequency Hz`:	Serial communication clock frequency for synchronous communication (CSI and CSIHS protocols), expressed in Hertz.

Description:

This commands sets the serial communication clock frequency when using the CSI and CSIHS protocols.

When using the UART protocol, this command is not necessary.

TCSETPAR FOSC

Command syntax:

```
TCSETPAR FOSC <frequency Hz>
```

Parameters:

`frequency Hz`:	Target device’s oscillator frequency.

Description:

Sets the target device’s internal or external oscillator frequency.

TCSETPAR 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 ISP 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.

TCSETPAR VDD_AUX

Command syntax:

```
TCSETPAR VDD_AUX <voltage mV>
```

Parameters:

voltage mV:	Auxiliary supply voltage, expressed in millivolts, in the range 3000-14500mV.

Description:

This command is used to generate an optional, auxiliary voltage level for user purposes. The specified voltage is routed to the VPROG1 line of the FlashRunner “ISP” connector.

A value of 0 drives the VPROG1 line to GND. If the `TCSETPAR VDD_AUX` is not sent, the VPROG1 line is driven to HiZ.

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.

TCSETPAR RSTUP

Command syntax:

```
TCSETPAR RSTUP <time μs>
```

Parameters:

time μs:	Reset rising time, expressed in microseconds.

Description:

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 is not long enough, FlashRunner could not be able to enter the programming mode.

TCSETPAR RSTDOWN

Command syntax:

```
TCSETPAR RSTDOWN <time μs>
```

Parameters:

time μs:	Reset falling time, expressed in microseconds.

Description:

The Reset rising 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. If RSTDOWN is not long enough, FlashRunner could not be able to enter the programming mode.

TCSETPAR RSTDRV

Command syntax:

```
TCSETPAR RSTDRV PUSHPULL|OPENDRAIN
```

Parameters:

`PUSHPULL`:	The Reset line is driven in push-pull mode.
`OPENDRAIN`:	The Reset line is driven in open drain mode.

Description:

Drives the Reset line in push-pull or open drain mode. If the **`TCSETPAR RSTDRV`** command is not sent, FlashRunner drives the Reset line in open drain mode by default.

TCSETPAR CLKOUT

Command syntax:

```
CSETPAR CLKOUT <frequency Hz>
```

Command options:

`frequency Hz`:	Frequency of the clock signal to be generated at the CLKOUT pin of the FlashRunner “ISP” connector, expressed in Hertz.

Description:

Generates an auxiliary clock signal at the CLKOUT pin of the FlashRunner “ISP” connector. This signal can be used as an auxiliary clock source, and is particularly useful when the target microcontroller requires an external clock that is not otherwise available on the target board.

This command is mandatory (and the CLKOUT line of FlashRunner must be connected to the target device) when programming 78K0S Kx1+ devices.

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

Note: *due to the FlashRunner internal circuitry and other factors, the frequency generated by FlashRunner is usually not the exact frequency specified by the* ***`TCSETPAR CLKOUT`*** *command. However, as a response to the* ***`TCSETPAR CLKOUT`*** *command, FlashRunner answers with the actual frequency that will be used.*

4. 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.

NEC 78K-specific target programming commands are the following:

- **`TPCMD BLANKCHECK`**;
- **`TPCMD MASSERASE`**;
- **`TPCMD BLOCKERASE`**;
- **`TPCMD PROGRAM`**;
- **`TPCMD VERIFY`**;
- **`TPCMD PROTECT`**;
- **`TPCMD RUN`**.

TPCMD BLANKCHECK

Command syntax:

```
TPCMD BLANKCHECK <tgt start addr> <len>
```

Command options and parameters:

`tgt start address`:	Device memory location from where the blankcheck operation will start.
`len`:	Number of locations to be blankchecked.

Description:

Blankchecks Flash memory. Blankchecks **`len`** locations starting from the address specified by **`tgt start address`**.

`tgt start address` must be the first location of a block and **`len`** must be a multiple of block length.

TPCMD MASSERASE

Command syntax:

`TPCMD MASSERASE`

Description:

Mass erases Flash memory.

TPCMD BLOCKERASE

Command syntax:

`TPCMD BLOCKERASE <tgt start addr> <len>`

Command options and parameters:

`tgt start address`:	Device memory location from where the block erase operation will start.
`len`:	Number of locations to be erased.

Description:

Erases one or more Flash memory blocks.

`tgt start address` must be the first location of a block and **`len`** must be a multiple of block length.

TPCMD PROGRAM

Command syntax:
`TPCMD PROGRAM <src offset> <tgt start addr> <len>`

Command options and parameters:

`src offset`:	Offset from the beginning of the source memory.
`tgt start addr`:	Device memory location from where the program operation will start.
`len`:	Number of locations to be programmed.

Description:

Programs **`len`** locations in the Flash memory starting from the **`tgt start addr`** address.

`tgt start address` must be the first location of a block.

On single-voltage devices, if **`len`** is not a multiple of block length, remaining locations in the last affected block are filled with 0xFF.

On dual-voltage devices, if **`len`** is not a multiple of 256, additional locations are filled with 0xFF until a number of locations which is a multiple of 256 is reached. In other words, if **`len`** is not a multiple of 256, 256 - (**`len`** mod 256) additional locations are filled with 0xFF.

Note: *the* ***`TPCMD PROGRAM`*** *command returns an error if the device is not blank. Therefore, the* ***`TPCMD BLANKCHECK`*** *command is not necessary.*

TPCMD VERIFY

Command syntax:

```
TPCMD VERIFY <src offset> <tgt start addr> <len>
```

Command options and parameters:

`src offset`:	Offset from the beginning of the source memory.
`tgt start addr`:	Device memory location from where the verify operation will start.
`len`:	Number of locations to be verified.

Description:

Verifies **`len`** locations in the Flash memory starting from the **`tgt start addr`** address.

`tgt start address` must be the first location of a block.

On single-voltage devices, if **`len`** is not a multiple of block length, remaining locations in the last affected block are verified against 0xFF.

On dual-voltage devices, if **`len`** is not a multiple of 256, additional locations are verified against 0xFF until a number of locations which is a multiple of 256 is reached. In other words, if **`len`** is not a multiple of 256, 256 - (**`len`** mod 256) additional locations are verified against 0xFF.

TPCMD PROTECT

Command syntax:

```
TPCMD PROTECT <security byte>
TPCMD PROTECT <security byte> <fsws> <fswe>
```

Command options and parameters:

`security byte`:	Security byte.
`fsws`:	Flash shield window start block number (for 78K0R devices only).
`fswe`:	Flash shield window end block number (for 78K0R devices only).

Description:

Programs the security byte. This operation is not supported by dual voltage devices.

TPCMD RUN

Command syntax:

`TPCMD RUN`

Command parameters:

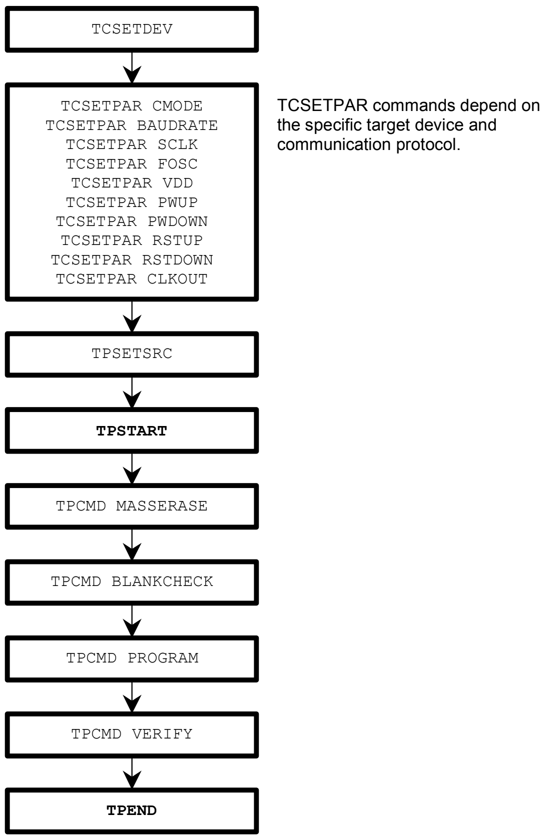
None.

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

The example provided below will help you understand how the commands discussed above should be used for a typical 78K device, in CSI communication mode. In this case, the device is a UPD78F0515.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR NEC UPD78F0515
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
;
; DIO0 (FLMD0)
; DIO1 (Not used)
; DIO2 (TxD) Device Output
; DIO3 (RxD) Device Input
; DIO4 (SCK)
; DIO5 (Not used)
; DIO6 (RESET)
; CLKOUT (Not used)
;

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

; Sets device
TCSETDEV NEC UPD78F0515 NEC78K

;------------------------
;FLASHRUNNER I/O Settings
;------------------------

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

; Clock oscillator frequency driven by FlashRunner, Hz
; Possible frequencies are: 25000000 divided by a 16-bit prescaler, 0 (DISABLED)
TCSETPAR CLKOUT 0

; VDD rise-time, ms (from 0 ms to 65535 ms)
TCSETPAR PWUP 10

; VDD fall-time, ms (from 0 ms to 65535 ms)
TCSETPAR PWDOWN 10

; RESET rise-time, us (from 0 us to 65535 us)
TCSETPAR RSTUP 100

; RESET fall-time, us (from 0 us to 65535 us)
TCSETPAR RSTDOWN 100

;--------------------
;NEC78K ALGO Settings
;--------------------

; Communication mode settings (UART or CSI supported)
TCSETPAR CMODE CSI

; External clock source frequency, Hz (change as needed)
; For this device, in CSI communication mode, the FOSC is fixed to 8000000 Hz
TCSETPAR FOSC 8000000

; Serial clock settings, Hz (change as needed)
; The CSI communication mode needs to have the serial clock set.
; For this device the maximum SCLK is 2500000 Hz.
TCSETPAR SCLK 2500000

;---------------------------
;Start Programming operation
;---------------------------

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

; Starts programming block
TPSTART

;--------------
;FLASH commands
;--------------

; Mass erases Flash memory
TPCMD MASSERASE

; Blank checks Flash memory (change address and length as needed)
TPCMD BLANKCHECK $0 $F000

; Programs Flash memory (change addresses and length as needed)
TPCMD PROGRAM $0 $0 $F000

; Verifies Flash memory (change addresses and length as needed)
TPCMD VERIFY $0 $0 $F000

; Ends programming block
TPEND
```

The example provided below will help you understand how the commands discussed above should be used for a typical 78K device, in UART 1-wire communication mode. In this case, the device is a UPD78F9211.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR NEC UPD78F9211
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
;
; DIO0 (Not used)
; DIO1 (Not used)
; DIO2 (DGDATA)
; DIO3 (Not used)
; DIO4 (Not used)
; DIO5 (Not used)
; DIO6 (RESET)
; CLKOUT (DGCLK)
;

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

; Sets device
TCSETDEV NEC UPD78F9211 NEC78K

;------------------------
;FLASHRUNNER I/O Settings
;------------------------

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

; Clock oscillator frequency driven by FlashRunner, Hz
; Possible frequencies are: 25000000 divided by a 16-bit prescaler, 0 (DISABLED)
TCSETPAR CLKOUT 8000000

; VDD rise-time, ms (from 0 ms to 65535 ms)
TCSETPAR PWUP 10

; VDD fall-time, ms (from 0 ms to 65535 ms)
TCSETPAR PWDOWN 10

; RESET rise-time, us (from 0 us to 65535 us)
TCSETPAR RSTUP 100

; RESET fall-time, us (from 0 us to 65535 us)
TCSETPAR RSTDOWN 100

;--------------------
;NEC78K ALGO Settings
;--------------------

; Communication mode settings (only UART 1 wired supported)
TCSETPAR CMODE UART

; External clock source frequency, Hz (change as needed)
; For this device the typical FOSC value is 8000000 Hz.
; Optional values are 6000000, 9000000 or 10000000 Hz.
TCSETPAR FOSC 8000000

;--------------------------
;Start Programming operation
;--------------------------

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

; Starts programming block
TPSTART

;--------------
;FLASH commands
;--------------

; Mass erases Flash memory
TPCMD MASSERASE

; Blank checks Flash memory (change address and length as needed)
TPCMD BLANKCHECK $0 $800

; Programs Flash memory (change addresses and length as needed)
TPCMD PROGRAM $0 $0 $800

; Ends programming block
TPEND
```

The FlashRunner’s system software setup will install script examples specific for each device of the 78K family in your PC.

7. Programming Times

The following table shows programming times for selected NEC 78K devices.

Device	Memory Size	Comm. Mode	Osc. Freq.	Operations	Time
UPD78F0393	32KB Flash	CSI (2.5 MHz)	8 MHz	Program	2.37 s
UPD78F0393	32KB Flash	CSI (2.5 MHz)	8 MHz	Erase + Program + Verify	3.81 s
UPD78F0515	60KB Flash	CSI (2.5 MHz)	8 MHz	Program	4.12 s
UPD78F0515	60KB Flash	CSI (2.5 MHz)	8 MHz	Erase + Program + Verify	6.67 s
UPD78F0889	96KB Flash	CSI (2.5 MHz)	8 MHz	Program	6.69 s
UPD78F0889	96KB Flash	CSI (2.5 MHz)	8 MHz	Erase + Program + Verify	10.65 s
UPD78F9211	2KB Flash	UART1	6 MHz	Program	1.70 s
UPD78F9211	2KB Flash	UART1	6 MHz	Erase + Program	1.94 s

Programming times depend on Programming Algorithm version, target board connections, communication mode, 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

Microcontroller-specific datasheets