AN00145: Interfacing FlashRunner with Microchip PIC24, dsPIC30, dsPIC33 Devices

FlashRunner is a Universal In-System Programmer which uses the principles of In-Circuit Serial Programming (better know as ICSP) to program Microchip PIC microcontrollers. This application note describes how to properly set up and use FlashRunner to program PIC24, dsPIC30 and dsPIC33 Flash devices.

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

1. Introduction

In-system programming of PIC microcontrollers is performed by entering the device's program/verify mode, which allows the programming of the MCU memory, through the ICSP protocol. This protocol accepts serial data and clock on two of the port pins and writes the information to the appropriate target memory space.

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

SofTec™
Microsystems

Revision 1.0 – April 2008

DC10010

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

E-mail (technical support): support@smh-tech.com

Web: http://www.smh-tech.com

Important

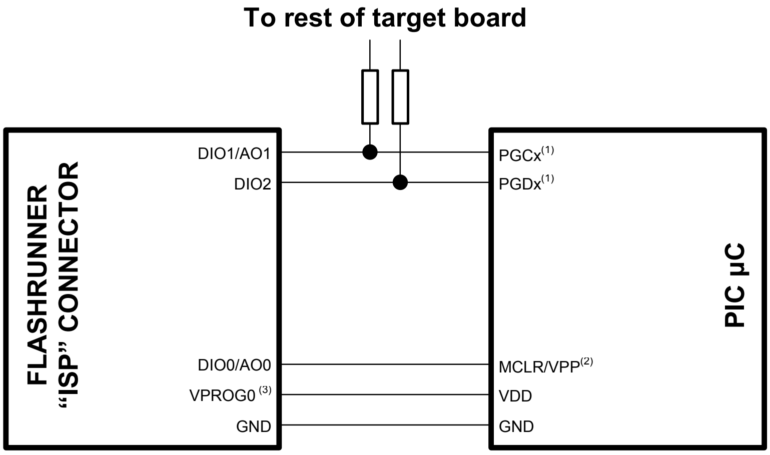
SMH Technologies reserves the right to make improvements to its products, their documentation and software routines, without notice. Information in this manual is intended to be accurate and reliable. However, SMH Technologies assumes no responsibility for its use; nor for any infringements of rights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

SMH TECHNOLOGIES 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.

2. Hardware Configuration

The microcontroller lines needed to implement the ICSP protocol and program a PIC device are the following:

- **MCLR:** Program Mode select.
- **VDD:** Device power supply voltage. This line must be connected if you want FlashRunner to automatically power the target device.
- **PGDx (or DATA):** This pin is used for data input/output during serial operation.
- **PGCx (or CLOCK):** This pin is used for clock input during serial operation.
- **GND:** Device power supply ground.



Notes

[1] If the lines PGCx and PGDx needed to enter program/verify mode are used for other purposes in the application, series resistors 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.

[2] For dsPIC30: special care should be taken in isolating the MCLR/VPP line from the rest of your target board because, in high-voltage programming mode, this signal could have a voltage level about 13V—which could damage the target board if the necessary measures are not taken. Note also that large capacitive loads on the MCLR/VPP line could increase the signal rise time, preventing the MCU from entering the Program/Verify mode.

[3] Connect this line if you want FlashRunner to power the target board.

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.

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

- $V_{DD;}$
- Power up time;
- Power down time;
- Serial clock frequency;
- Programming voltage (only for dsPIC30 devices).

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 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 Program/Verify 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 Program/Verify 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 SCLK

Command syntax:

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

Parameters:

frequency Hz: serial communication clock frequency, expressed in Hertz.

Description:

This commands sets the serial communication clock frequency. When powering the device with a voltage in the low part of the working range or when the serial communication lines have high capacitive loads, the serial communication clock frequency should be lowered in order to avoid communication errors.

TCSETPAR VPP

Command syntax:

```
TCSETPAR VPP <voltage mV>
```

Parameters:

voltage mV: Target device programming voltage, expressed in millivolts. Typically in the 10-13V range.

Description:

This command is only necessary for dsPIC30 devices, which need a High-Voltage Programming (HVP) mode, and is used to properly generate the appropriate programming voltage in the MCLR/VPP line.

If not specified, a default programming high voltage of about 11V is automatically generated by FlashRunner.

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.

The PIC18-specific target programming commands are the following:

- **TPCMD BLANKCHECK**;
- **TPCMD ERASE**;
- **TPCMD MASSERASE**;
- **TPCMD PROGRAM**;
- **TPCMD VERIFY**;
- **TPCMD READ**;
- **TPCMD RUN**.

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 memory and **E** stands for EEPROM memory.
tgt start address:	Device memory location from where the blankcheck operation will start.
len:	Number of locations to be blankchecked.

Description:

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

Flash memory is organized as 24-bit locations: **len** specifies the number of 24-bit words to be blankchecked.

EEPROM memory is organized as 16-bit locations: **len** specifies the number of 16-bit words to be blankchecked.

TPCMD ERASE

Command syntax:

```
TPCMD ERASE F|E <tgt row start addr> <len>
```

Command options and parameters:

F|E:	Memory type to be mass erased (Flash or EEPROM).
tgt row start addr:	Row start address.
len:	Length of the memory block to mass erase.

Description:

Erases Flash (program memory) or EEPROM memory.

Erases one or more memory rows. The **tgt row start addr** parameter must be the first location in a row, and **len** must be a multiple of a row length. Note: this command only works with devices that support row erasing.

Flash memory is organized as 24-bit locations: **len** specifies the number of 24-bit words to be erased.

EEPROM memory is organized as 16-bit locations: **len** specifies the number of 16-bit words to be erased.

TPCMD MASSERASE

Command syntax:

```
TPCMD MASSERASE C
```

Command options:

None.

Description:

Performs a complete mass erase (program memory, EEPROM and configuration words). This command is also needed for mass erasing protected devices.

TPCMD PROGRAM

Command syntax:

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

Command options and parameters:

`F`:	Flash memory (program memory and configuration words).
`E`:	EEPROM memory.
`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 or EEPROM memory starting from the **`tgt start addr`** address.

Please note that program memory and configuration words (both in Flash) must be programmed separately.

Flash memory is organized as 24-bit locations: **`len`** specifies the number of 24-bit words to be programmed.

EEPROM memory is organized as 16-bit locations: **`len`** specifies the number of 16-bit words to be programmed.

Note for Devices NOT Belonging to the PIC24F Series

Please note that an Intel-Hex file generated by Microchip development tools contains both Flash memory (including configuration words) and EEPROM memory. Every memory zone to be programmed to the target device must be separately converted to a FlashRunner binary file.

Conversion is done by using the ***fr_bin2frb*** *utility, installed in the local in the local* ***\UTILITIES*** *folder. The syntax of the* ***fr_bin2frb*** *utility is the following:*

```
fr_bin2frb  -i <hex file> -t hex -s <start addr> -e <end addr> -u FF -o
            <output file>
```

where ***<start addr>*** *and* ***<end addr>*** *are the start and end address, respectively, of the memory zone under consideration, as logically mapped in the hex file. Please note that, while the Intel-Hex files generated by Microchip development tools addresses increment by one every byte, on the target device memory addresses increment by two every 24-bit (for Flash memory) or 16-bit word (for EEPROM memory). The Intel-Hex file zero-pads every 24-bit Flash value or 16-bit EEPROM value into a 32-bit value. Consequently, the* ***<start addr>*** *and* ***<end addr>*** *parameters are obtained by multiplying by two the target device's physical addresses. Specific batch files for every target device are provided which automatically convert an Intel-Hex file into the corresponding FlashRunner binary files for every device's memory zone.*

The following examples show how to correctly generate FlashRunner binary files for a dsPIC30F4013 device.

Example of program memory code conversion:

```
fr_bin2frb -i source.hex -t hex -s 0 -e FFFF -u FF -o flash.frb
```

Example of configuration words conversion:

```
fr_bin2frb -i source.hex -t hex -s 1F00000 -e 1F0001B -u FF -o cw.frb
```

Example of EEPROM conversion:

```
fr_bin2frb -i source.hex -t hex -s FFF800 -e FFFFFF -u FF -o eeprom.frb
```

The portion of script that programs and verifies all of the three memory zones will look like the following:

```
; Programs and verifies Flash memory
TPSETSRC FILE FLASH.FRB
TPCMD PROGRAM F $0 $0 16384
TPCMD VERIFY F $0 $0 16384

; Programs and verifies EEPROM memory
TPSETSRC FILE EEPROM.FRB
TPCMD PROGRAM E $0 $7FFC00 512
TPCMD VERIFY E $0 $7FFC00 512

; Programs and verifies configuration words
TPSETSRC FILE CW.FRB
TPCMD PROGRAM F $0 $F80000 7
TPCMD VERIFY F $0 $F80000 7
```

TPCMD VERIFY

Command syntax:

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

Command options and parameters:

`F`:	Flash memory (program memory, user ID locations and configuration words).
`E`:	EEPROM memory.
`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 or EEPROM memory starting from the **`tgt start addr`** address.

Please note that program memory and configuration words (both in Flash) must be verified separately.

Flash memory is organized as 24-bit locations: **`len`** specifies the number of 24-bit words to be verified.

EEPROM memory is organized as 16-bit locations: **`len`** specifies the number of 16-bit words to be verified.

TPCMD READ

Command syntax:

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

Command parameters and options:

`F|E`:	Specifies Flash (**`F`**) or EEPROM (**`E`**) memory.
`tgt start addr`:	Device memory location from where the read operation will start.
`len`:	Number of locations to be read.

Description:

Reads **`len`** locations of Flash or EEPROM memory starting from the **`tgt start addr`** address.

Flash memory is organized as 24-bit locations: **`len`** specifies the number of 24-bit words to be read.

EEPROM memory is organized as 16-bit locations: **`len`** specifies the number of 16-bit words to be read.

TPCMD RUN

Command syntax:

```
TPCMD RUN
```

Command parameters:

None.

Description:

Runs the target application.

5. Script Examples

The examples provided below will help you understand how the commands discussed above should be used for three typical PIC devices: a PIC24FJ64GA010 device, a dsPIC30F4013 and a dsPIC33FJ64GP710 device.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR MICROCHIP PIC24FJ64GA010
;
; Use this example as a starting point for your specific programming needs
;
; ---------------------
; HARDWARE CONNECTIONS
; ---------------------
; DIO0 (MCLR)
; DIO1 (PGC)
; DIO2 (PGD)
;

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

; Sets device
TCSETDEV MICROCHIP PIC24FJ64GA010 MCHP_A

;----------
; SETTINGS
;----------
; Oscillator frequency, Hz (change as needed)
TCSETPAR SCLK 5000000

; Power Supply Voltage, mV (change as needed)
TCSETPAR VDD 3300

; VDD rise-time, ms (change as needed)
TCSETPAR PWUP 1

; VDD fall-time, ms (change as needed)
TCSETPAR PWDOWN 1

;---------------------------
; START PROGRAMMING SESSION
;---------------------------
TPSTART

; Erases FLASH, EEPROM and Configuration Registers also if code protected
TPCMD MASSERASE C

; Blank checks Flash Program memory (length is expressed in 24-bit words)
TPCMD BLANKCHECK F $0 22014

; Blank checks Flash Configuration Registers (change addresses and length as needed)
TPCMD BLANKCHECK F $ABFC 2

; Image file to be programmed (must be placed in the \BINARIES directory)
; This File has been generated using the FR_BIN2FRB conversion utility (see PIC24FJ64GA010.bat)
TPSETSRC FILE FLASH.FRB

; Programs Flash Program Memory (length is expressed in 24-bit words)
TPCMD PROGRAM F $0 $0 22014

; Verifies Flash Program Memory (length is expressed in 24-bit words)
TPCMD VERIFY F $0 $0 22014

; Programs Configuration Registers (length is expressed in 24-bit words)
TPCMD PROGRAM F $ABFC $ABFC 2

; Verifies Configuration Registers (length is expressed in 24-bit words)
TPCMD VERIFY F $ABFC $ABFC 2

; Ends programming session
TPEND
```

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR MICROCHIP DSPIC30F4013
;
; Use this example as a starting point for your specific programming needs
;
; ---------------------
; HARDWARE CONNECTIONS
; ---------------------
; DIO0 (MCLR)
; DIO1 (PGC)
; DIO2 (PGD)
;

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

; Sets device
TCSETDEV MICROCHIP DSPIC30F4013 MCHP_A

;----------
; SETTINGS
;----------
; Oscillator frequency, Hz (change as needed)
TCSETPAR SCLK 5000000

; Power Supply Voltage, mV (change as needed)
TCSETPAR VDD 5000

; VDD rise-time, ms (change as needed)
TCSETPAR PWUP 1

; VDD fall-time, ms (change as needed)
TCSETPAR PWDOWN 1

;---------------------------
; START PROGRAMMING SESSION
;---------------------------
TPSTART

; Erases FLASH, EEPROM and Configuration Registers also if code protected
TPCMD MASSERASE C

; Blank checks Flash Program memory (length is expressed in 24-bit words)
TPCMD BLANKCHECK F $0 16384

; Blank checks Flash Configuration Registers (change addresses and length as needed)
TPCMD BLANKCHECK F $F80000 7

; Blank Checks EEPROM memory (length is expressed in 16-bit words)
TPCMD BLANKCHECK E $7FFC00 512

; Image file to be programmed (must be placed in the \BINARIES directory)
; This File has been generated using the FR_BIN2FRB conversion utility (see DSPIC30F4013.bat)
TPSETSRC FILE FLASH.FRB

; Programs Flash Program Memory (length is expressed in 24-bit words)
TPCMD PROGRAM F $0 $0 16384

; Verifies Flash Program Memory (length is expressed in 24-bit words)
TPCMD VERIFY F $0 $0 16384

; Image file to be programmed into the EEPROM(must be placed in the \BINARIES directory)
; This File has been generated using the FR_BIN2FRB conversion utility (see DSPIC30F4013.bat)
TPSETSRC FILE EEPROM.FRB

; Programs EEPROM memory (length is expressed in 16-bit words)
TPCMD PROGRAM E $0 $7FFC00 512

; Verifies EEPROM memory (length is expressed in 16-bit words)
TPCMD VERIFY E $0 $7FFC00 512

; Image file to be programmed into the Configuration Registers (must be placed in the \BINARIES directory)
; This File has been generated using the FR_BIN2FRB conversion utility (see DSPIC30F4013.bat)
TPSETSRC FILE CW.FRB

; Programs Configuration Registers (length is expressed in 16-bit words)
TPCMD PROGRAM F $0 $F80000 7

; Verifies Configuration Registers (length is expressed in 16-bit words)
TPCMD VERIFY F $0 $F80000 7

; Ends programming session
TPEND
```

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR MICROCHIP DSPIC33FJ64GP710
;
; Use this example as a starting point for your specific programming needs
;
; ---------------------
; HARDWARE CONNECTIONS
; ---------------------
; DIO0 (MCLR)
; DIO1 (PGC)
; DIO2 (PGD)
;

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

; Sets device
TCSETDEV MICROCHIP DSPIC33FJ64GP710 MCHP_A

;----------
; SETTINGS
;----------
; Oscillator frequency, Hz (change as needed)
TCSETPAR SCLK 5000000

; Power Supply Voltage, mV (change as needed)
TCSETPAR VDD 3300

; VDD rise-time, ms (change as needed)
TCSETPAR PWUP 1

; VDD fall-time, ms (change as needed)
TCSETPAR PWDOWN 1

;---------------------------
; START PROGRAMMING SESSION
;---------------------------
TPSTART

; Erases FLASH, EEPROM and Configuration Registers also if code protected
TPCMD MASSERASE C

; Blank checks Flash Program memory (length is expressed in 24-bit words)
TPCMD BLANKCHECK F $0 22016

; Blank checks Flash Configuration Registers (change addresses and length as needed)
TPCMD BLANKCHECK F $F80000 12

; Image file to be programmed (must be placed in the \BINARIES directory)
; This File has been generated using the FR_BIN2FRB conversion utility (see DSPIC33FJ64GP710.bat)
TPSETSRC FILE FLASH.FRB

; Programs Flash Program Memory (length is expressed in 24-bit words)
TPCMD PROGRAM F $0 $0 22016

; Verifies Flash Program Memory (length is expressed in 24-bit words)
TPCMD VERIFY F $0 $0 22016

; Image file to be programmed into the Configuration Registers (must be placed in the \BINARIES directory)
; This File has been generated using the FR_BIN2FRB conversion utility (see DSPIC33FJ64GP710.bat)
TPSETSRC FILE CW.FRB

; Programs Configuration Registers (length is expressed in 8-bit words)
TPCMD PROGRAM F $0 $F80000 12

; Verifies Configuration Registers (length is expressed in 8-bit words)
TPCMD VERIFY F $0 $F80000 12

; Ends programming session
TPEND
```

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

6. Programming Times

The following table shows programming times for selected Microchip PIC family devices.

Device	Mem. Size	Conditions	Operations	Time
PIC24FJ64GA002	64KB Flash	SCLK = 4.16MHz	Erase/Blank Check/Program/Verify	5.00 s
dsPIC30F4013	48KB Flash	SCLK = 4.16MHz	Erase/Blank Check/Program/Verify	5.88 s
dsPIC33FJ128MC706	128KB Flash	SCLK = 4.16MHz	Erase/Blank Check/Program/Verify	14.34 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. SMH Technologies reserves the right to modify Programming Algorithms at any time.

7. References

FlashRunner user’s manual

Microcontroller-specific datasheets