

AN00138: Interfacing FlashRunner with Microchip PIC10/12/16 Devices

by Mirco Ambroset (mirco.ambroset@softecmicro.com)

FlashRunner is a Universal In-System Programmer which uses the principles of In-Circuit Serial Programming (better known as ICSP) to program Microchip PIC microcontrollers. This application note describes how to properly set up and use FlashRunner to program PIC10, PIC12 and PIC16 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

Microchip PIC10, PIC12 and PIC16 devices can be divided into two subfamilies based on the core architecture:

- **Baseline Core Devices:** these devices feature a 12-bit wide code memory, and a two-level call stack. They are represented by PIC10 series, as well as some PIC12 and PIC16 devices.
- **Mid-Range Core Devices:** these devices feature a 14-bit wide code memory, and an improved eight-level call stack. The instruction set differs very little from the baseline devices, but the increased opcode width allows more memory to be directly addressed. The mid-range core is available in the majority of devices labelled PIC12 and PIC16.

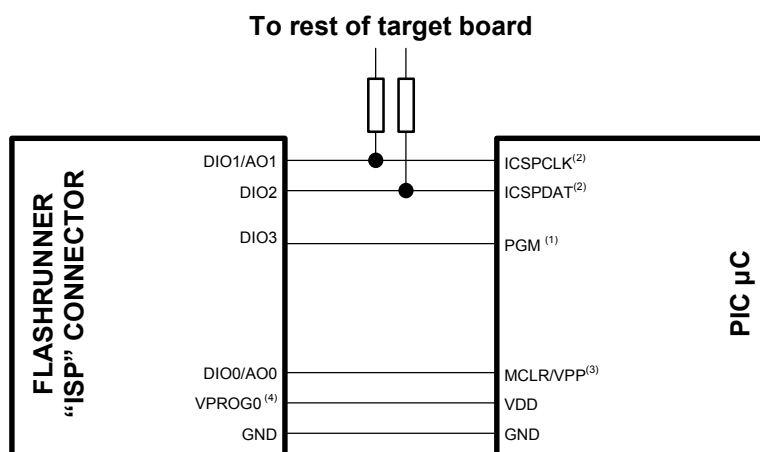
In-system programming of PIC microcontrollers is performed by entering the device's program/verify mode, which allows the programming of the MCU memory, through the ICSP protocol. This protocol accepts serial data and clock on two of the port pins and writes the information to the appropriate target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

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

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. The range of programming voltage is typically 4.5–5.5V, depending on the MCU. This line must be connected if you want FlashRunner to automatically power the target device.
- **ICSPDAT (or DATA):** This pin is used for data input/output during serial operation.
- **GND:** Device power supply ground.
- **ICSPCLK (or CLOCK):** This pin is used for clock input during serial operation.
- **PGM:** This pin must be connected only for devices that feature the Low Voltage Programming (LVP) mode.



Notes

- (1) Connect this line only if you choose to enter program/verify mode with low voltage programming mode.
- (2) If the lines ICSPDAT and ICSPCLK 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.
- (3) 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.

⁽⁴⁾ Connect this line if you want FlashRunner to power the target board (recommended).

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 mode;
- Programming voltage (only for high-voltage programming mode).

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 monitor 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 command sets the serial communication clock frequency. This frequency should not exceed the device's internal RC clock frequency, which is typically 4MHz.

When powering the device with a low voltage (less than 4.5V) 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 PROGMODE

Command syntax:

```
TCSETPAR PROGMODE HVP|LVP
```

Command options:

HVP: uses the High-Voltage Programming mode.

LVP: uses the Low-Voltage Programming mode.

Description:

Sets the programming mode.

The High-Voltage Programming (HVP) mode is supported by all devices. Some devices also support the Low-Voltage Programming (LVP) mode.

When using the HVP mode, a high programming voltage (typically in the range 10-13V) must be supplied to the device's MCLR/VPP pin.

When using the LVP mode, no high voltage needs to be supplied to the MCLR/VPP pin, but an additional line (PGM) must be connected to FlashRunner. Additionally, the LVP mode is only available if the LVP bit in the configuration word is set to 1 (default value for blank devices).

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 needed for the High-Voltage Programming (HVP) mode, and is used to properly generate the appropriate programming voltage in the MCLR/VPP line.

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 PIC10/12/16-specific target programming commands are the following:

- **TPCMD BLANKCHECK;**
- **TPCMD MASSERASE;**
- **TPCMD PROGRAM;**
- **TPCMD VERIFY;**
- **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**.

TPCMD MASSERASE

Command syntax:

```
TPCMD MASSERASE F|E|C
```

Command options:

F|E|C: Memory type to be mass erased (Flash, EEPROM, Complete).

Description:

Mass erases Flash memory, EEPROM memory, or both.

TPCMD MASSERASE F mass erases the Flash memory (program memory).

TPCMD MASSERASE E mass erases the EEPROM memory.

TPCMD MASSERASE C performs a complete mass erase (program memory, EEPROM and configuration memory). **TPCMD MASSERASE C** is also needed for mass erasing protected devices.

TPCMD PROGRAM

Command syntax:

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

Command options and parameters:

F:	Flash memory (program memory and configuration memory).
E:	EEPROM memory.
B W:	EEPROM Addressing mode (Byte or Word).
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 memory (both in Flash) must be programmed separately.

The EEPROM addressing mode specifies the addressing mode of the source memory. Use **w** when using Intel-Hex files generated by Microchip development tools, since these files use two bytes (the first being 0x00) for every EEPROM byte. Use **B** when using a byte-aligned memory source.

Please note that, on Intel-Hex files generated by Microchip development tools, EEPROM locations are logically mapped starting from address 0x2100. In this case, a typical **TPCMD PROGRAM** command will look like the following:

```
TPCMD PROGRAM E W $2100 $0 128
```

Also note that, on baseline core devices (PIC10 series, as well as some PIC12 and PIC16 devices), the Configuration Word register is stored at the logical address location of 0xFFF within the hex file. This logical address location may not reflect the actual physical address for the part itself. As an example, in the PIC12F508, the Configuration Word is located at the 0x3FF address. In this case, a typical **TPCMD PROGRAM** command to program the Configuration Word will look like the following:

```
TPCMD PROGRAM F $FFF $3FF 1
```

In any case, the script examples installed by the FlashRunner's system software setup are specific for each device of the PIC family and contain the correct addresses and parameters for programming the specific part.

TPCMD VERIFY

Command syntax:

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

Command options and parameters:

F:	Flash memory (program memory and configuration memory).
E:	EEPROM memory.
B W:	EEPROM Addressing mode (Byte or Word).
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 memory (both in Flash) must be verified separately.

The EEPROM addressing mode specifies the addressing mode of the source memory. Use **w** when using Intel-Hex files generated by Microchip development tools, since these files use two bytes (the first being 0x00) for every EEPROM byte. Use **B** when using a byte-aligned memory source.

TPCMD RUN

Command syntax:

TPCMD RUN

Command parameters:

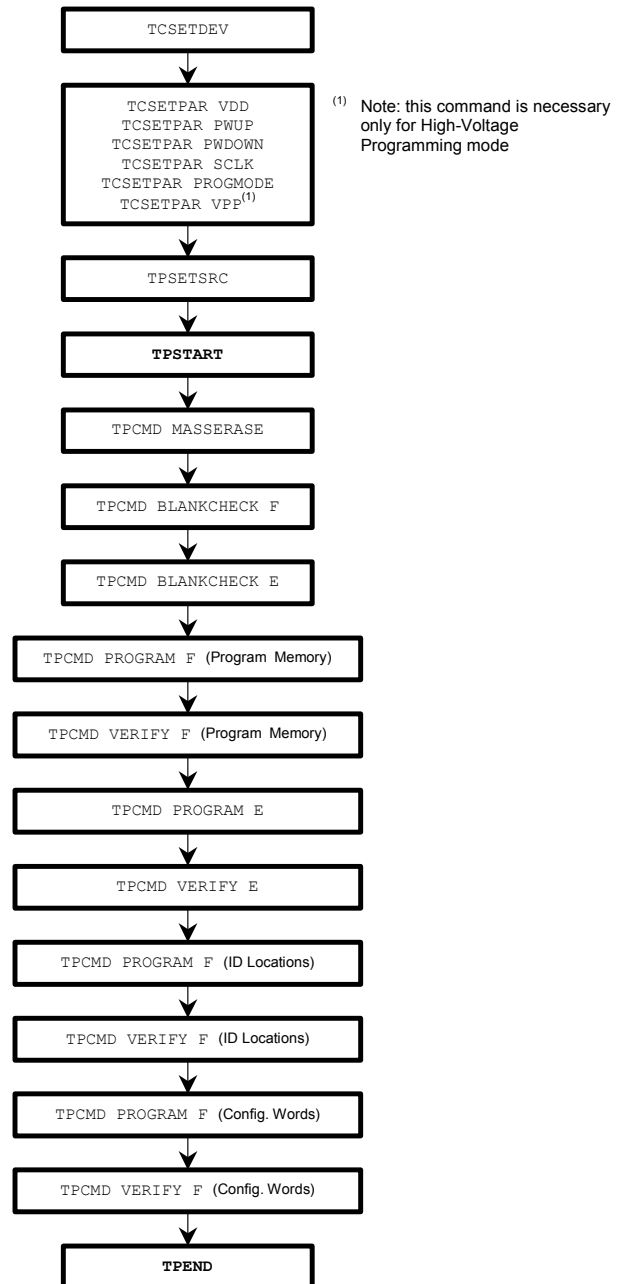
None.

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 Example

The example provided below will help you understand how the commands discussed above should be used for a typical PIC device, in this case the PIC16F628A.

```

;
; FLASHRUNNER SCRIPT EXAMPLE FOR MICROCHIP PIC16F628A
;
; Use this example as a starting point for your specific programming needs
;
; -----
;
; Hardware connections
;
; DIO0 (MCLR)
; DIO1 (ICSPCLK)
; DIO2 (ICSPDAT)
; DIO3 (PGM) only for Low-Voltage Programming mode
;

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

; Sets device
TCSETDEV MICROCHIP PIC16F628A PIC16

; Serial clock frequency, Hz (change as needed)
TCSETPAR SCLK 4000000

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

; Sets High-Voltage Programming mode
TCSETPAR PROGMODE HVP

; Programming voltage, mV (change as needed)
TCSETPAR VPP 13000

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

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

; Starts programming block
TPSTART

; Erases FLASH, EEPROM, Configuration Word and UserId Locations even if code protected
TPCMD MASSERASE C

; Blank checks Flash memory, UserId and Configuration Word (change address and length as needed)
TPCMD BLANKCHECK F $0 8200

; Blank Checks EEPROM memory (change address and length as needed)
TPCMD BLANKCHECK E $0 128

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

; Programs Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $0 $0 2048

; Verifies Flash memory (change addresses and length as needed)
TPCMD VERIFY F $0 $0 2048

; Programs EEPROM memory from the same source of the Flash memory (Word addressing mode)
TPCMD PROGRAM E W $2100 $0 128

; Verifies EEPROM memory
TPCMD VERIFY E W $2100 $0 128

; Programs UserId locations
TPCMD PROGRAM F $2000 $2000 4

; Verifies UserId locations
TPCMD VERIFY F $2000 $2000 4

; Programs Configuration Word
TPCMD PROGRAM F $2007 $2007 1

; Verifies Configuration Word
TPCMD VERIFY F $2007 $2007 1

; Ends programming session
TPEND

```

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

7. Programming Times

The following table shows programming times for several devices representative of the PIC10/12/16 families.

Device	Memory Size	Comm. Frequency	Operations	Time
PIC10F222	512 W Flash	4 MHz	Program	1.18 s
PIC10F222	512 W Flash	4 MHz	Erase + Program + Verify	1.25 s
PIC12F675	1 KW Flash	4 MHz	Program	2.80 s
PIC12F675	1 KW Flash	4 MHz	Erase + Program + Verify	2.86 s
PIC16F628A	2 KW Flash	4 MHz	Program	5.30 s
PIC16F628A	2 KW Flash	4 MHz	Erase + Program + Verify	5.35 s
PIC16F873A	4 KW Flash	4 MHz	Program	0.70 s
PIC16F873A	4 KW Flash	4 MHz	Erase + Program + Verify	1.10 s
PIC16F88	4 KW Flash	4 MHz	Program	2.20 s
PIC16F88	4 KW Flash	4 MHz	Erase + Program + Verify	2.60 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
Microcontroller-specific datasheets