AN00144: Interfacing FlashRunner with Atmel AVR8 Devices

FlashRunner is a Universal In-System Programmer, which uses the principles of In-Circuit Programming to program Atmel AVR8 family microcontrollers.

This Application Note assumes that you are familiar with both FlashRunner and the main features of the AVR8 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 AVR8 microcontrollers is performed through a synchronous serial protocol (SPI).

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

SofTec™ Microsystems

Copyright © 2008 SMH Technologies, S.r.l.

SMH Technologies is the licensee of the SofTec Microsystems trademark.

Revision 1.0 – April 2008

DC10009

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.

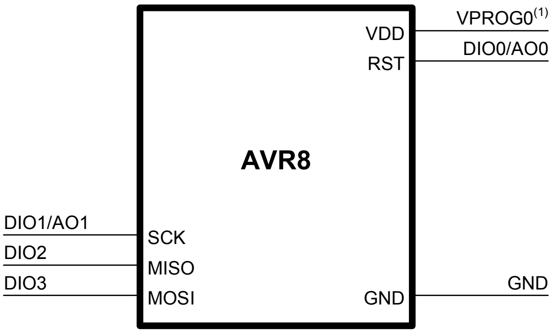
SMH TECHNOLOGIES WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

2. Hardware Configuration

The microcontroller lines needed to program an AVR8 device are the following:

- **SCK:** SPI serial clock.
- **MISO:** Master-In Slave-Out.
- **MOSI:** Master-Out Slave-In.
- **RST:** Device reset input/output pin.
- **VDD:** Device power supply voltage.
- **GND:** Device power supply ground.

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



[1] Connect this line if you want FlashRunner to automatically power the target device

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 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};
- CPU 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 FCPU

Command syntax:

`TCSETPAR FCPU <frequency Hz>`

Parameters:

`frequency Hz`: CPU frequency, expressed in Hertz.

Description:

Specifies the target microcontroller's frequency, according to what specified by fuse settings. The CPU frequency is used to automatically generate the SPI clock, using the following formula:

$$f_{SCK} = \begin{cases} \frac{f_{CPU}}{4} & (\text{if } f_{CPU} < 12\,\text{MHz}) \\ \frac{f_{CPU}}{6} & (\text{if } f_{CPU} \geq 12\,\text{MHz}) \end{cases}$$

Typically, the fuse settings of a blank device specify a CPU frequency of 1MHz.

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.

Atmel AVR8-specific target programming commands are the following:

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

- `TPCMD READ`;
- `TPCMD SAVECALIBRVALUE`;
- `TPCMD RUN`.

TPCMD BLANKCHECK

Command syntax:

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

Command parameters and options:

`F|E`: Specifies Flash (`F`) or EEPROM (`E`) 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 16-bit locations: `len` specifies the number of 16-bit words to be blankchecked.

EEPROM memory is organized as 8-bit locations: `len` specifies the number of bytes to be blankchecked.

TPCMD MASSERASE

Command syntax:

```
TPCMD MASSERASE
```

Command options:

None.

Description:

Mass erases Flash memory, EEPROM memory, and lock bits. Fuse bits are not erased. EEPROM memory will not be mass erased if the EESAVE fuse bit is set.

TPCMD PROGRAM

Command syntax:

```
TPCMD PROGRAM F|E <src offset> <tgt start addr> <len>
TPCMD PROGRAM FLB|FHB|EFB <fuse byte>
TPCMD PROGRAM LKB <lock byte>
```

Command parameters and options:

`F|E`: Specifies Flash (`F`) or EEPROM (`E`) 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.

`FLB|FHB|EFB`: Specifies the fuse byte to be programmed (Fuse Low Bits, Fuse High Bits or Extended Fuse Bits).

`fuse byte`: Fuse byte value to be programmed.

`LKB`: Programs Lock Bits.

`lock byte`: Lock byte value to be programmed.

Description:

Programs `len` locations of Flash or EEPROM memory starting from the `tgt start addr` address, or program the specified Fuse Bits or Lock Bits.

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

EEPROM memory is organized as 8-bit locations: `len` specifies the number of bytes to be programmed.

TPCMD VERIFY

Command syntax:

```
TPCMD VERIFY F|E <src offset> <tgt start addr> <len>
TPCMD VERIFY FLB|FHB|EFB <fuse byte>
TPCMD VERIFY LKB <lock byte>
```

Command parameters and options:

`F|E`: Specifies Flash (`F`) or EEPROM (`E`) 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.

`FLB|FHB|EFB`: Specifies the fuse byte to be verified (Fuse Low Bits, Fuse High Bits or Extended Fuse Bits).

`fuse byte`: Fuse byte value to be verified.

`LKB`: Verifies Lock Bits.

`lock byte`: Lock byte value to be verified.

Description:

Verifies **`len`** locations of Flash or EEPROM memory starting from the **`tgt start addr`** address, or verifies the specified Fuse Bits or Lock Bits.

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

EEPROM memory is organized as 8-bit locations: **`len`** specifies the number of bytes to be verified.

TPCMD READ

Command syntax:

```
TPCMD READ F|E <tgt start addr> <len>
TPCMD READ FLB|FHB|EFB|LKB
```

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.

`FLB|FHB|EFB|LKB`: Specifies the fuse byte or lock byte to be read (Fuse Low Bits, Fuse High Bits, Extended Fuse Bits or Lock Bits).

Description:

Reads **`len`** locations of Flash or EEPROM memory starting from the **`tgt start addr`** address, or reads the specified Fuse Bits or Lock Bits.

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

EEPROM memory is organized as 8-bit locations: **`len`** specifies the number of bytes to be read.

TPCMD SAVECALIBRVALUE

Command syntax:

```
TPCMD SAVECALIBRVALUE F|E <addr> <frequency Hz>
```

Command parameters and options:

`F|E`: Specifies Flash (**`F`**) or EEPROM (**`E`**) memory.

`addr`: Device memory location where to store the calibration value.

`frequency Hz`: Frequency, expressed in Hertz, of which to retrieve the corresponding calibration value.

Description:

Retrieves the factory programmed calibration value (for the internal RC oscillator) for the specified frequency and saves it to the specified Flash or EEPROM location.

TPCMD RUN

Command syntax:

```
TPCMD RUN
```

Command parameters:

None.

Description:

Runs the target application.

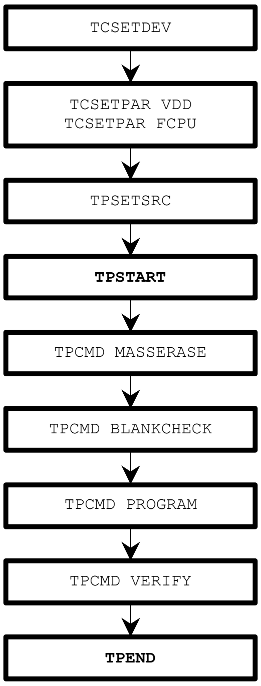
Speeding up programming

Since a blank device typically works with an internal RC oscillator frequency of 1MHz, this results in a slow SPI communication speed. To speed up programming, it is suggested to program the Low Fuse Bits so that they specify an higher internal RC oscillator frequency, typically 8MHz. The following script example illustrates this technique.

```
;***********************************
;First Programming block (FCPU=1MHz)
;***********************************

; Sets the VDD Voltage(mV)
TCSETPAR VDD 5000

; Sets the CPU clock frequency(Hz) (default value 1 MHz)
TCSETPAR FCPU  1000000

; Starts first programming block
TPSTART

;-------------------
;FUSE BYTES commands
;-------------------

; Programs Fuse Low Bits in order to set internal RC to 8 MHz
TPCMD PROGRAM FLB $E2

; Verifies Fuse Low Bits
TPCMD VERIFY FLB $E2

; Ends first programming block
TPEND

;************************************
;Second Programming block (FCPU=8MHz)
;************************************

; Sets the VDD Voltage(mV)
TCSETPAR VDD 5000

; Now sets the CPU clock frequency (Hz) to 8 MHz according to fuse settings
TCSETPAR FCPU  8000000

; Starts second programming block
TPSTART

...

PROGRAMMING OPERATIONS

...

; Ends second programming block
TPEND
```

5. Typical Programming Flow

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



6. Script Example

The example below shows a typical programming flow for an Atmel ATMega16 device.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR ATMEL ATMEGA16
;
; Use this example as a starting point for your specific programming needs
;
; ---------------------
; HARDWARE CONNECTIONS
; ---------------------
; DIO0 (RST)
; DIO1 (SCLK)
; DIO2 (MISO)
; DIO3 (MOSI)
;

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

; Sets device
TCSETDEV ATMEL ATMEGA16 AVR8

;----------
; SETTINGS
;----------
; Oscillator frequency, Hz (change as needed)
TCSETPAR FCPU 1000000

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

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

; Mass erases Flash memory and EEProm (only if high_fuse_bit EESAVE=1)
TPCMD MASSERASE

; Blank checks Flash memory (address and lenght are expressed in 16 bit Words)
TPCMD BLANKCHECK F $0 8192

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

; Programs Flash memory (address and lenght are expressed in 16 bit Words)
TPCMD PROGRAM F $0 $0 8192

; Verifies Flash memory, read-out method (address and lenght are expressed in 16 bit Words)
TPCMD VERIFY F $0 $0 8192

; Blank Checks EEPROM memory (address and lenght are expressed in Bytes)
TPCMD BLANKCHECK E $0 512

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

; Erases and Programs EEPROM memory (address and lenght are expressed in Bytes)
TPCMD PROGRAM E $0 $0 512

; Verifies EEPROM memory, read-out method (address and lenght are expressed in Bytes)
TPCMD VERIFY E $0 $0 512

; Programs Fuse Bits (fuses changes have no effect until the programming session ends,
; fuses are rewritable but cannot be erased by a MASSERASE)

; Programs Fuse Low Bits
TPCMD PROGRAM FLB $E1

; Verifies Fuse Low Bits
TPCMD VERIFY FLB $E1

; Programs Fuse High Bits
TPCMD PROGRAM FHB $99

; Verifies Fuse High Bits
TPCMD VERIFY FHB $99

; Ends programming block
TPEND
```

The FlashRunner’s system software setup will install script examples specific for each device of the AVR8 family on your PC.

7. Programming Times

The following table shows programming times for selected Atmel AVR8 family devices.

Device	Mem. Size	Conditions	Operations	Time
ATMEGA16	8KW Flash	fCPU=8MHz	Erase + Program + Verify	1.90 s
ATMEGA128	64KW Flash	fCPU=8MHz	Erase + Program + Verify	11.50 s
ATTINY861	4KW Flash	fCPU=8MHz	Erase + Program + Verify	1.40 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.

8. References

FlashRunner user's manual

Microcontroller-specific datasheets