AN00147: Using FlashRunner Power Supply Features to Power the BUP

By Loris Pacquola (loris.pacquola@smh-tech.com)

1. Introduction

The purpose of this Application Note is to help the user to understand and profitably use the power supplying capabilities of FlashRunner Series on In-System Programmers.

In-System Programming refers to programming the non-volatile memory of a device mounted on an electronic board. This means that appropriate powering must be supplied to the entire board (or a part of it) in order to achieve correct programming. The applied voltage should be within the range specified by the board designer, and care must be taken in order to guarantee that the required current is supplied. In case of multiple power supplies, the various power sources must be provided with the correct timing.

Despite the fact that power supplying is a fundamental aspect of any electronic system, sometimes not all of the issues related to it are taken into account, resulting in some undesired effects such as power voltage dropping below the minimum recommended level, undesired Reset events generated by LVD peripherals, or diffused noise coupled on power lines. For these reasons, a deep analysis of the board under programming should be performed.

SMH Technologies

Revision 1.0 – July 2008

DC10037

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

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

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

2. ISP Powering Issues

Special care must be taken when powering a Flash device during In-System Programming. Flash devices feature an embedded charge pump for generating the high voltage needed to charge the floating gate of the cell transistor. This simplifies the ISP interface, since a separate VPP (programming voltage) is not needed, but requires higher accuracy on the supplied voltage.

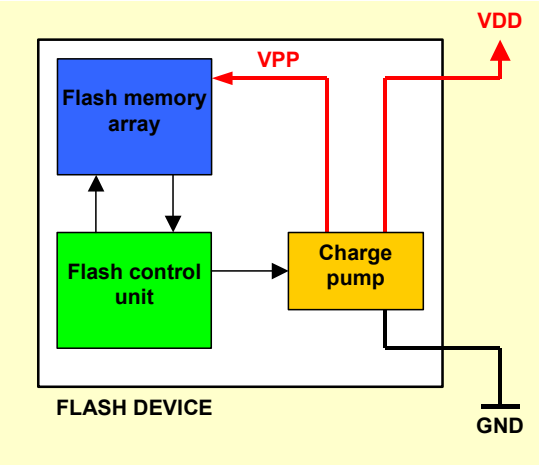

Figure 1. Flash Memory Programming Voltage

The direct relation between the Flash device power supply and the high voltage applied during cell programming prompted the request for power voltage monitoring during in-system programming. Correct powering during in-system programming is needed in order to achieve the maximum memory data retention declared by semiconductor manufacturers.

3. Typical Powering Schemes

Different powering schemes for the Board Under Programming (BUP) are possible. The paragraphs below illustrate some typical powering scenarios.

Scheme A: Single, Direct Power Supply

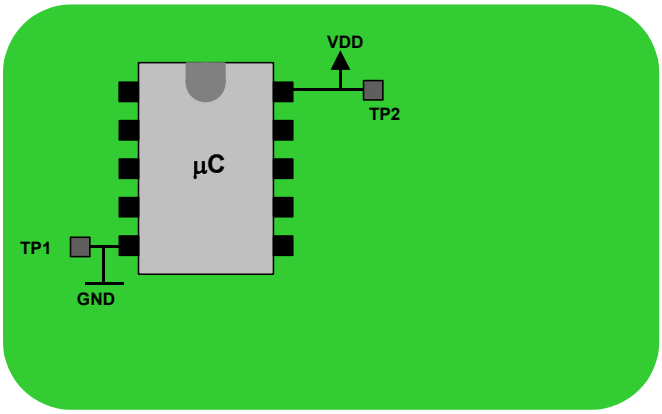

Figure 2. Single, Direct Power Supply

The device to be programmed is powered through the power pins. Care must be taken when this method is used, since other circuitry mounted on the board could be unpowered and high current paths could arise through the shared lines. In this case, the suggested powering method is to connect the FlashRunner VPROG0 power line to VDD and FlashRunner GND line to GND. The VPROG0 voltage must be set to the board nominal “VDD” voltage by the command:

```
TCSETPAR VDD <target_voltage [mV]>
```

Scheme B: Dual, Direct Power Supply

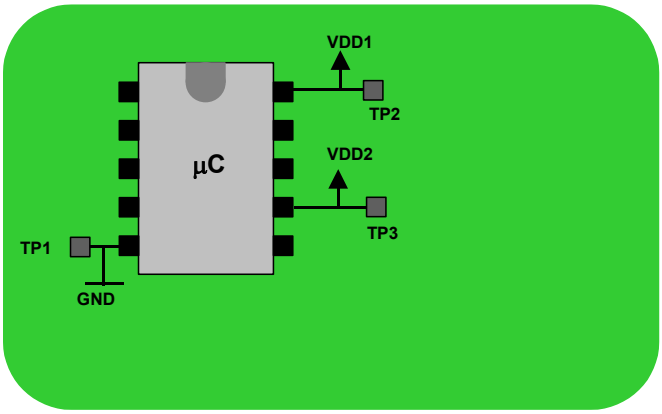

Figure 2. Dual, Direct Power Supply

The device to be programmed is powered through two or more power lines. This is the typical case of sub-system boards, where the board is a daughter board of a more complex system and the power stage is located on the main board. Different voltages and timings may be needed for “VDD1” and “VDD2” power lines, and this can be achieved thanks to FlashRunner VPROG0 and VPROG1 power lines through the commands:

```
TCSETPAR VDD  <target_voltage 1 [mV]>
TCSETPAR VDD_AUX <target_voltage 2 [mV]>
```

Scheme C: Single Power Supply

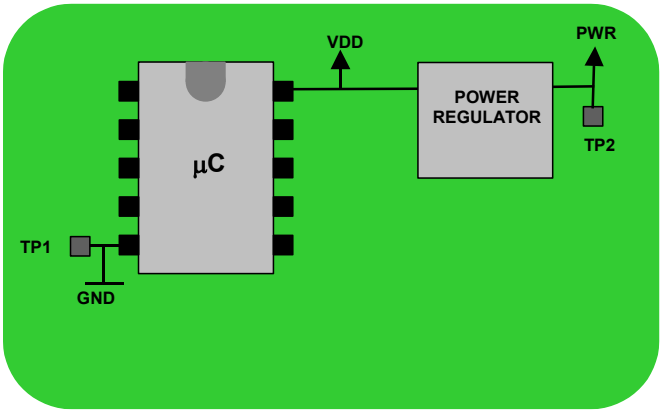

Figure 3. Single Power Supply

The device to be programmed is powered through the on-board power regulator. In this case it is not suggested, during the programming phase, to power the board directly through the VDD and GND lines: in fact, the rest of the circuitry will stay unpowered or powered below the nominal range. This situation could generate a high current path through the protection input stage of unpowered devices. Additionally, sometimes the low output impedance of the power regulator could draw a lot of current from the programmer output power line. In addition, unpowered devices mounted on the board through the ISP shared lines could generate some communication errors: the low impedance of the unpowered device will pull down the ISP signal, especially in case of open drain communication.

Scheme D: Dual Power Supply

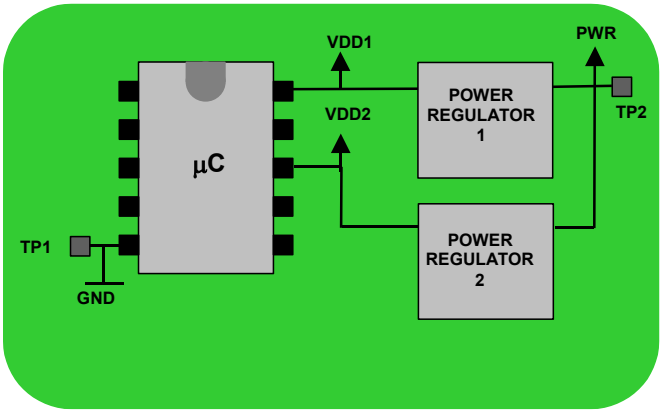

Figure 4. Dual Power Supply

The device to be programmed is powered thanks to two or more on-board power supplies. As in the previous scenario, it is not suggested to

power the device directly through VDD1 and VDD2 power lines: it's better to feed the entire board using the main power line "PWR".

4. Timing, Voltage and Current Considerations

The voltage required to program a BUP depends on the devices mounted, the application design and other issues. Usually, during the programming phase, the power lines should be driven at normal working levels; however, some microcontrollers need a greater or lower voltage in order to perform special memory verify operations. The voltage applied to the DUP (Device Under Programming) must always be specified in the script file even if the board is powered through an external power supply. In fact, all of the ISP lines are driven to the voltage specified by the command:

```
TCSETPAR VDD <target_voltage [mV]>
```

The block diagram below illustrates FlashRunner's Power and ISP stages.

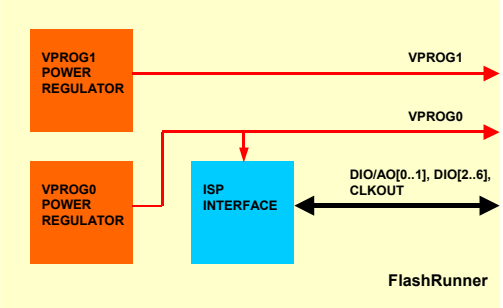

Figure 5. FlashRunner Power and ISP Interface

There are two programmable power regulators, VPROG1 and VPROG0, and one "ISP INTERFACE" block, powered through VPROG0. This means that care must be taken when two power supplies are required. In this case, VPROG0 must be used to power the DUP power lines that condition the DUP I/O lines: this will ensure that FlashRunner ISP lines and device's I/O lines (involved during the programming) match the same voltage levels.

Moreover, the current drawn by the BUP should be taken into account, in order to avoid voltage dropping during programming (due to the intervention FlashRunner's current limiter).

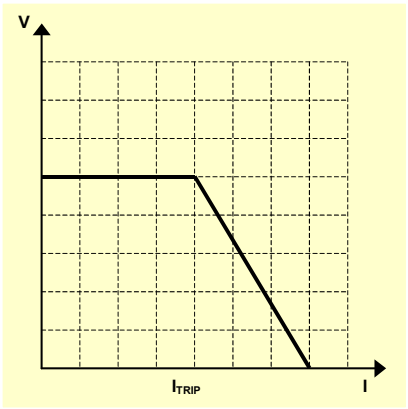

Figure 6. VPROG[0..1] V-I Characteristic

The maximum source current capability is 0.5A for VPROG0, and 0.25A for VPROG1, with a current constant regulation in case of overcurrent event.

Attention must be paid in case of high value capacitors on the BUP. Even if the nominal current is lower than the maximum current capability of FlashRunner's power lines, high current peaks may cause overvoltage events.

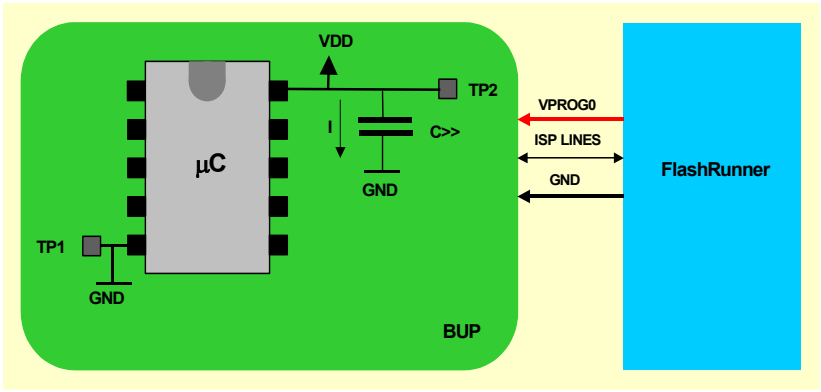

Figure 7. Interfacing Between BUP and FlashRunner

The phenomenon is emphasized because of the FlashRunner VPROG[0..1] driving method: the power voltage specified in the script is applied only after the nominal value is achieved inside the programmer, and this increases the total current amount requested:

$I_{TOT} = I_{NOM} + I_{CPEAK}$

Where I_{NOM} is the nominal current consumption during programming phase and I_{CPEAK} is:

$I_{CPEAK} = C \cdot dVc/dt$

This is the theoretical current request but, due to the current limiter with constant current regulation, the maximum current is limited to 0.5A (for VPROG0) and 0.25A (for VPROG1). The resulting voltage-time characteristic is shown in the diagram below.

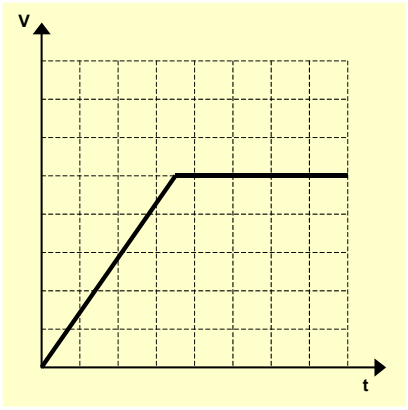

Figure 8. VPROG[0..1] Voltage-Time Characteristic

Timing

It is fundamental that programming starts after VDD has reached the nominal level, otherwise programming errors (or, in the worst case, data retention problems) could occur.

To do this, FlashRunner provides two specific commands:

```
TCSETPAR PWDOWN <value [ms]>
TCSETPAR PWUP <value [ms]>
```

The first command specifies the delay time between the VPROG[0..1] voltage is applied and the first operation performed on ISP lines; this acts like a wait state after the board power on.

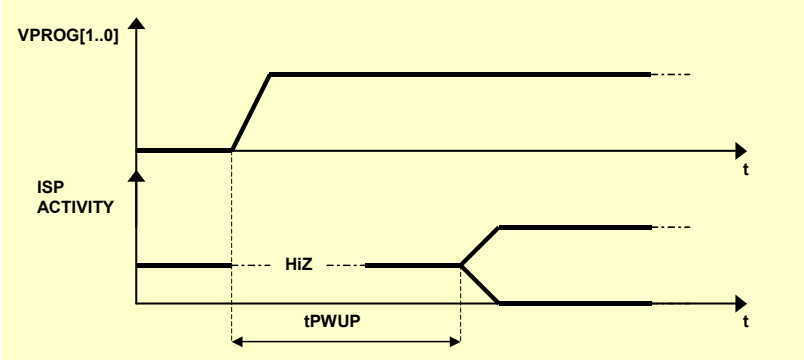

Figure 9. PWUP Parameter

To override delays caused by high value capacitors mounted on the BUP, a long enough PWUP time should be specified: usually, a safe value is the time needed for the VDD to reach its nominal value raised by 10%:

$t_{PWUP} = V_{DD\ RISEUP(100\%)} + 10\%\ V_{DD\ RISEUP(100\%)}$

The valid range for PWUP parameter is 0 to 65535 ms.

At the end of a script execution, FlashRunner drives the VPROG0 and VPROG1 lines to low impedance for a t_{PWDOWN} time. This will ensure a fast discharge of BUP's power lines, required both to speed-up the power off/on sequence needed for some microcontrollers during programming and to guarantee a known initial state of the BUP for successive in circuit tests.

The internal FlashRunner circuitry features a high-current, controlled shunt resistor for VPROG0 and VPROG1:

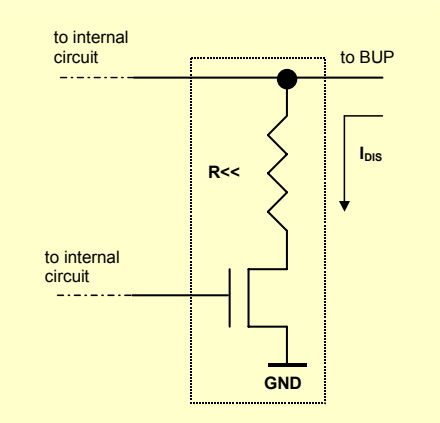

Figure 10. FlashRunner Discharge Circuitry

The resistor is automatically controlled by the FlashRunner's operating system every time the TPEND command is executed (even if an error has been encountered during script execution). The time spent with the shunt resistor enabled after the TPEND command execution (or every time a power OFF than ON sequence is required by the algorithm), is t_{PWDOWN}. After that, the VPROG0 and VPROG1 lines are driven to high impedance (HiZ).

The t_{PWDOWN} parameter should be set to the discharge time plus a safety margin of 10%.

$t_{PWDOWN} = t_{VDD\ FALLDOWN(100\%)} + 10\%\ t_{VDDFALLDOWN(100\%)}$

The valid range for PWDOWN parameter is 0 to 65535 ms.

To find out the optimum value, start from the default value (10 ms), measure with an oscilloscope the discharge time, and trim the new value. The default value usually fits the most common capacitor filters.

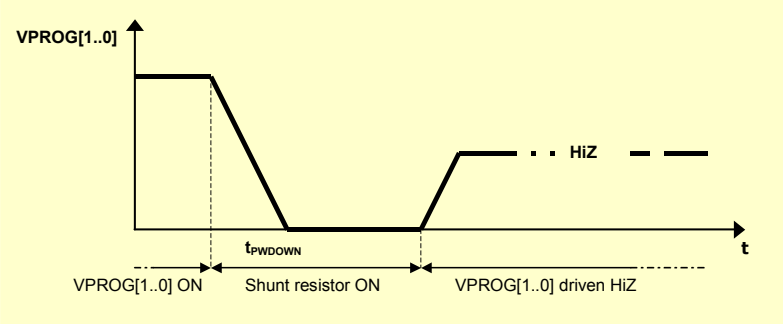

Figure 11. PWDOWN Parameter

5. Practical Application

This section will show a practical demonstration of In-System Programming with special focus on BUP powering.

The figure below shows a part of the BUP schematic. The power supply stage is composed of two power regulators with 5V output (linear regulator U202) and one programmable voltage output (3.3V or 5V) called “VDD” (buck regulator). All of the power lines can be contacted through test points.

The microcontroller is a MC9S12XEP100 device, a member of the Freescale HCS12X family.

The powering method chosen will provide a 12V voltage at TP3 and TP2 test points: in this way the entire BUP will be powered. The TP3 test point will allow for a fast discharge time since the block diode is bypassed. The capacitor mounted is a 47uF, polarized tantalum capacitor with a 100nF ceramic capacitor in parallel. The relatively low equivalent capacitor value won’t be an issue. The BUP nominal power consumption is 200mA, therefore FlashRunner’s VPROG1 line will be suitable as a power supply during In-System Programming.

The J202 jumper will be set to the “5V” position during In-System Programming.

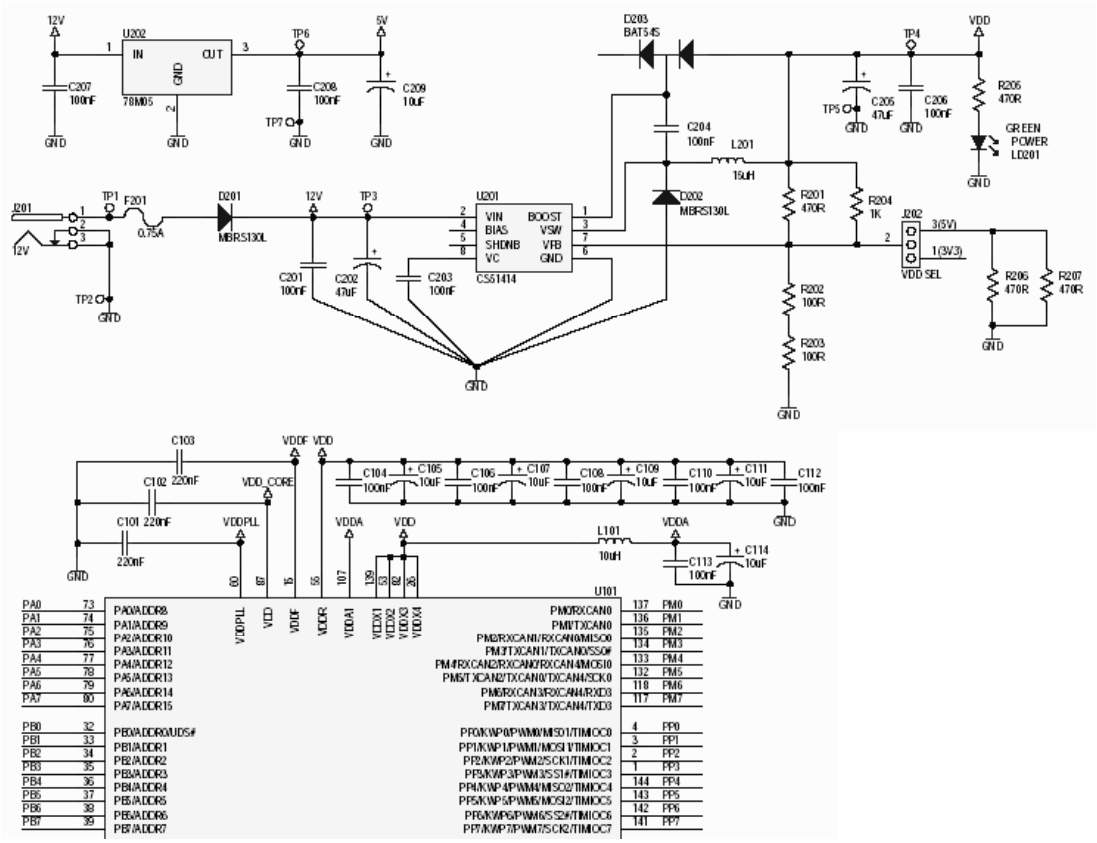

Figure 12. Part of Schematic of the BUP

Script Settings

The script used for board programming is the following.

```
;
; FLASHRUNNER SCRIPT FOR FREESCALE MC9S12XEP100
;
;
; ----------
;
; Hardware connections
;
; DIO0/AO0  (RESET)
; DIO2      (BKGD)
; CLKOUT    (CLOCK - optional)
;

; Turns off logging
#LOG_ON 2
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV FREESCALE MC9S12XEP100 HCS12

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

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

; VPROG1 voltage, mV (from 3000 to 14500, 0 to disable)(change as needed)
```

```
TCSETPAR VDD_AUX 12000

; Clock oscillator frequency driven by FlashRunner, Hz
; The possible frequencies are: 25000000, 12500000, 6250000, 0 (DISABLED)
TCSETPAR CLKOUT 0

; RESET down time (from 0 us to 65535 us)
TCSETPAR RSTDOWN 10
; RESET up time (from 0 us to 65535 us)
TCSETPAR RSTUP 100

; Power down time (from 0 ms to 65535 ms)
TCSETPAR PWDOWN 10
; Power up time (from 0 ms to 65535 ms)
TCSETPAR PWUP 10

;--------------------
;HCS12 ALGO Settings
;--------------------

; External clock source frequency, Hz (change as needed)
; For this device the maximum FOSC is 100000000 Hz
TCSETPAR FOSC 4000000

; Enables and sets the internal PLL clock frequency, Hz
; Uses the following formula to calculate the PLL parameters:
;      PLLFREQ = 2*FOSC*(SYNR+1)/(REFDIV+1)
; For this device the maximum PLLFREQ is 100000000 Hz
; Set all parameters to 0 to disable the internal PLL clock

TCSETPAR PLLFREQ 0
TCSETPAR REFDIV 0
TCSETPAR SYNR 0

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

; Starts programming block
TPSTART

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

; Chip unsecuring. Enable this command if the device is protected
; Note: This command erases Flash memory and EEPROM memory
TPCMD UNSECURE

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

; Mass erases Flash memory
TPCMD MASSERASE F

; Blank checks Flash memory (change address and length as needed)
; Paged FLASH: 0x8000 TO 0xBFFF addressed through PPAGE.
TPCMD BLANKCHECK F $C08000 $4000 P


; Programs Flash memory (change source and target address and length as needed)
; Paged FLASH: 0x8000 TO 0xBFFF addressed through PPAGE.
TPCMD PROGRAM F $C08000 $C08000 $4000 B P

; Verifies Flash memory, read-out method (change source and target address and length
as needed)
; Paged FLASH: 0x8000 TO 0xBFFF addressed through PPAGE.
TPCMD VERIFY F R $C08000 $C08000 $4000 B P

; Ends programming block
TPEND
```

The chosen VDD is 5V:

```
TCSETPAR VDD 5000
```

since, even if VPROG0 is not connected to the BUP, the ISP interface is powered to the VPROG0 value. This will ensure that the voltage levels of the ISP lines used (DIO0/AO0 and DIO2) will match the voltage levels of the MC9S12XEP100 RESET and BKGD lines.

VPROG1 is set to 12V:

```
TCSETPAR VDD_AUX 12000
```

and, as stated, it will be the main power supply for the BUP.

Finally, the power down and power up settings are left to their default value of 10ms:

```
TCSETPAR PWDOWN 10
TCSETPAR PWUP 10
```

Script Execution and Scope Captures

The figure below is a scope capture of the BUP's 12V power supply line driven by FlashRunner's VPROG1 programmable power supply. The time lapse between the 12V rising edge and the first activity on the BKGD line is 13ms, slightly longer than the specified PWUP parameter (10ms). This is normal, and due to the software overhead of the programming algorithm.

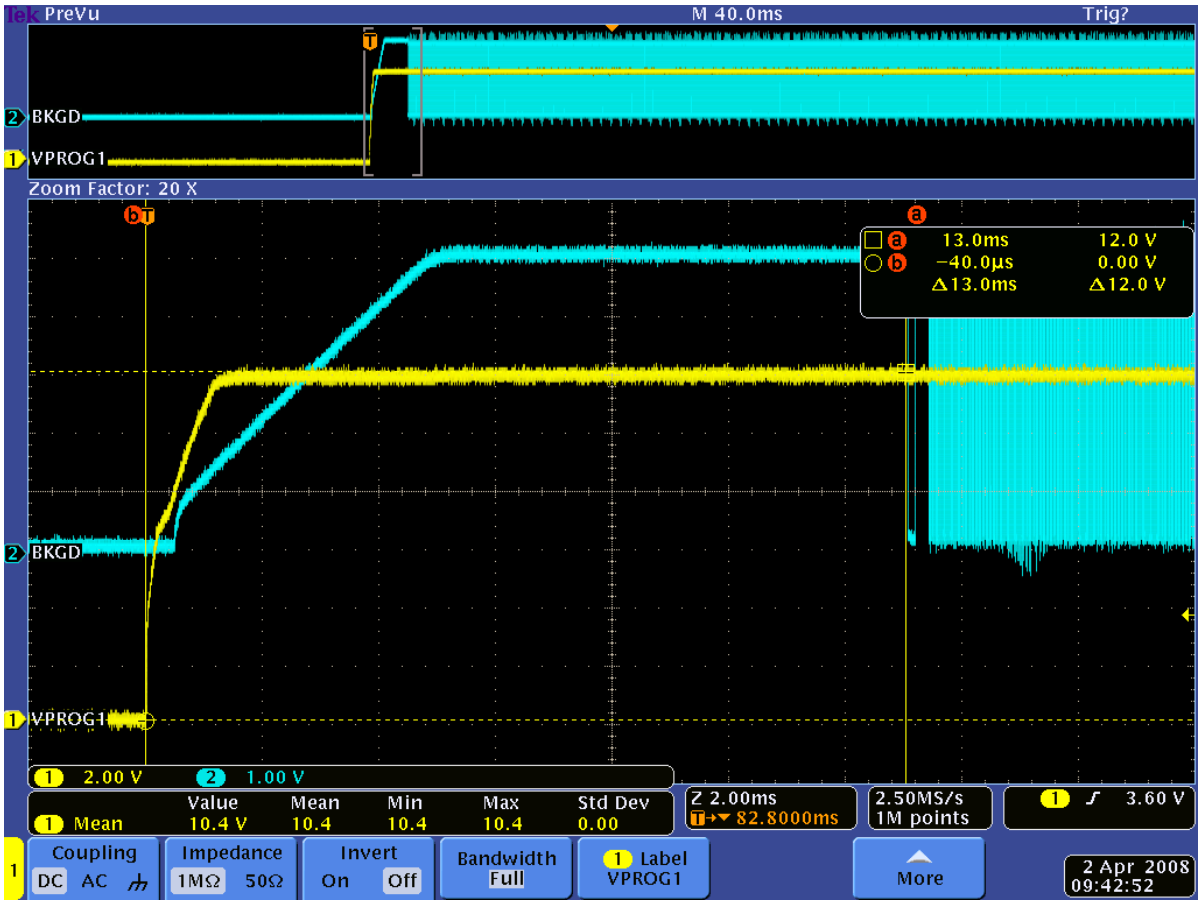

Figure 13. BUP Power On

The figure below shows the BUP power down sequence. The 12V power supply line is turned off in less than 9ms. This means that the specified PWDOWN parameter is large enough to guarantee a correct BUP power off.

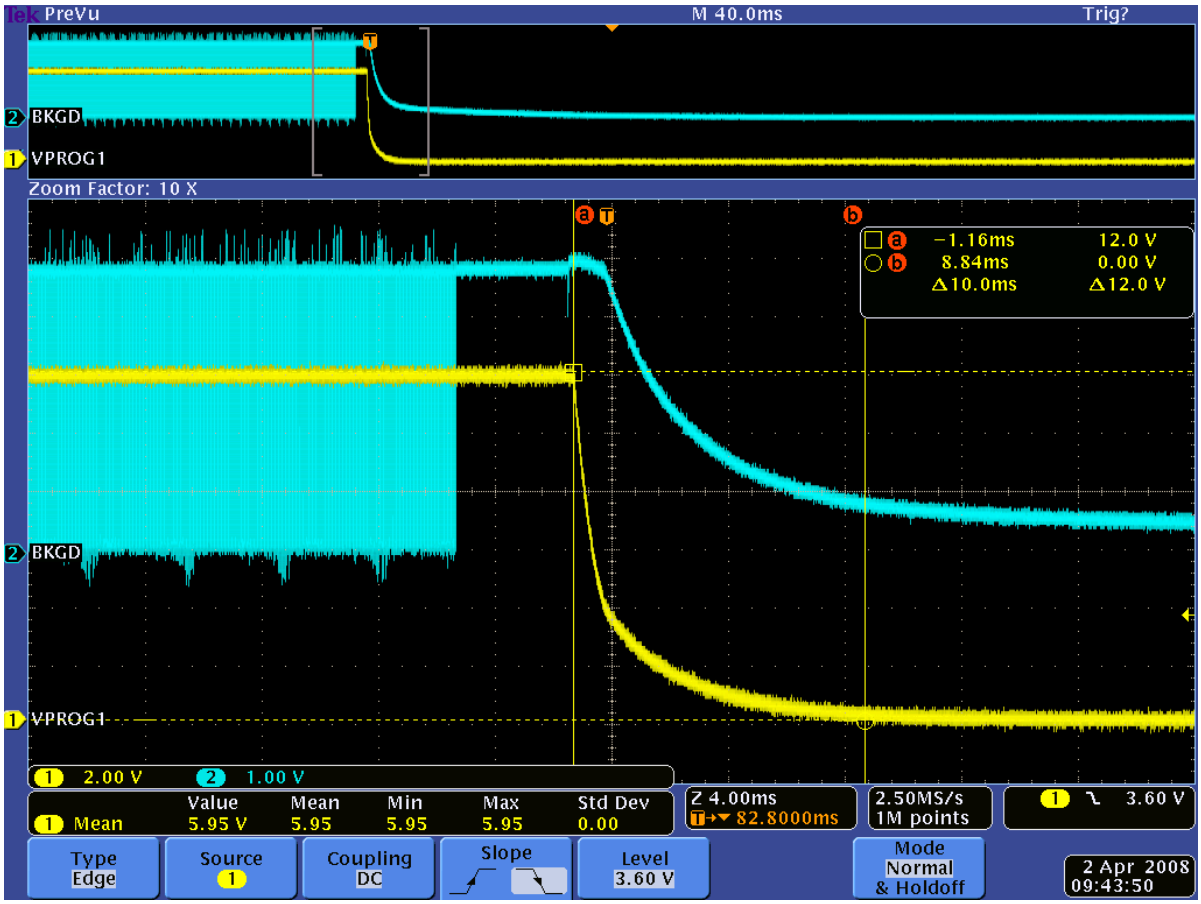

Figure 14. BUP Power Off

6. Conclusion

Board powering during In-System Programming should be careful evaluated through board schematic analysis, manufacturer nominal power supply values, and maximum current consumption rates. Then, using FlashRunner, the setting of the various parameters in the script will take into account the results of the previous analysis. Finally, some quick oscilloscope measures will tweak the chosen parameters, and will avoid undesired troubles during the production phase.