FLASHRUNNER

High-Performance, Standalone In-System Programmer

FR03
User's Manual

SMH Technologies™

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

FlashRunner FR03

High-Performance, Standalone In-System Programmer

User's Manual

Revision 1.3 - Apr 2015

SMH Technologies™

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

Our policy at SMH Technologies is to comply with all applicable worldwide safety and EMC/EMI regulations.
This product is certified to comply with the 2004/108/EC Directives and is in conformity with the EN6100-6-2 and the EN61000-6-3 standards.
In a domestic environment, this product may cause radio interference in which case the user may be required to take adequate prevention measures.
Attaching additional wiring to this product or modifying the product operation from the factory default as shipped may effect its performance and cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken.

Disposal of Waste Electrical & Electronic Equipment (WEEE).
In the European Union, this label indicates that this product should not be disposed of with household waste. It must be deposited in an appropriate facility to allow for recovery and recycling. For more detailed information about the recycling of this product, please contact your local city office, household waste disposal service or the retail store where you purchased this product.

SMH Technologies

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

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

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

Written by Paolo Xausa

Contents

Contents

Index of Figures

Index of Tables

0 Before Starting

> **i** **Note:** *the FlashRunner System Software CD-ROM and/or SMH Technologies website (www.smh-tech.com) may contain an updated version of this user's manual. Please check before continuing reading this documentation.*

0.1 Important Notice to Users

While every effort has been made to ensure the accuracy of all information in this document, SMH Technologies assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accidents, or any other cause.

0.2 Safety

FlashRunner is a low-voltage device. However, when integrating it inside an automatic test equipment or when interfacing it with other systems, take all precautions in order to avoid electrical shocks due to, for example, different ground references.

Make all connections to the target system before applying power to the instrument.

To protect FlashRunner against electrostatic discharge (ESD), always connect yourself to earth (e.g. via wrist straps) when handling the instrument.

Always store FlashRunner inside an antistatic bag when not in use.

0

0.3 Getting Technical Support

SMH Technologies is continuously working to improve FlashRunner firmware and to release programming algorithms for new devices. SMH Technologies offers a fast and knowledgeable technical support to all of its customers and is always available to solve specific problems or meet specific needs.

To get in touch with SMH Technologies, please refer to the contact information below.

Phone: +39 0434 421111

Fax: +39 0434 639021

Technical Support: *support@smh-tech.com*

0.4 Additional Documentation

This user's manual provides information about how to setup FlashRunner FR03 and its hardware characteristics.

For information about FlashRunner commands and their syntax, including specific commands for specific family of microcontrollers, please refer to the FlashRunner Programmer's Manual, included (in PDF format) in the FlashRunner CD-ROM.

1 Overview

1

1.1 What is FlashRunner FR03?

FlashRunner FR03 is a high-performance, standalone In-System Programmer specific for Flash-based microcontrollers and serial memories.

FlashRunner FR03 is targeted at production environments, easily interfaces to your programming system or Automatic Test Equipment (ATE) and can work either in full standalone mode or controlled by a host system.

Figure 1.1: FlashRunner FR03

1

1.1.1 General Features

- Fastest programming algorithms (as fast as target device's memory technology limit), approved by silicon manufacturers;
- Easy ATE integration;
- Standalone operations (projects and code images stored on a memory card);
- Also controllable by any host system via RS-232;
- Supports most ISP protocols (BDM, JTAG, SPI, I2C, MON, ICC, SCI, etc.);
- Flexible, fully configurable;
- Compact and robust design for production environments;
- Data integrity guaranteed (every data transfer to/from the host system or Secure Digital card is CRC tagged).

1.1.2 Hardware Features

- 7.5V DC power supply input;
- Five digital I/O lines;
- Two digital I/O or analog output lines;
- One programmable output voltage (0 to 5.5V, 0.5A);
- One programmable clock output;
- Secure Digital memory card (up to 2 GB);
- I/O protection;
- One command inputs (START);
- Three status outputs (FAIL, PASS, BUSY);
- RS-232 channel.

1.1.3 Software Features

- Fully autonomous standalone mode thanks to its SD memory card;
- Controllable by any host system through a terminal utility and simple ASCII protocol;
- One hardware-selectable project (script);
- Unlimited software-selectable projects (scripts);
- Erase, blank check, program, read, verify, oscillator trimming, etc.

1.2 Package Checklist

The FlashRunner FR03 package includes the following items:

- FlashRunner FR03 unit, including an SD card already pre-installed with the programming algorithm you specified at the time of purchase;
- Serial cable for connection to the host PC;
- FlashRunner "System Software" CD-ROM, containing the FlashRunner Control Panel utility and the FlashRunner Programmer's Manual in PDF format;
- This user's manual;
- Warranty card.

1.3 Hardware Overview

The figure below illustrates where the various FlashRunner FR03 connectors and controls are located.
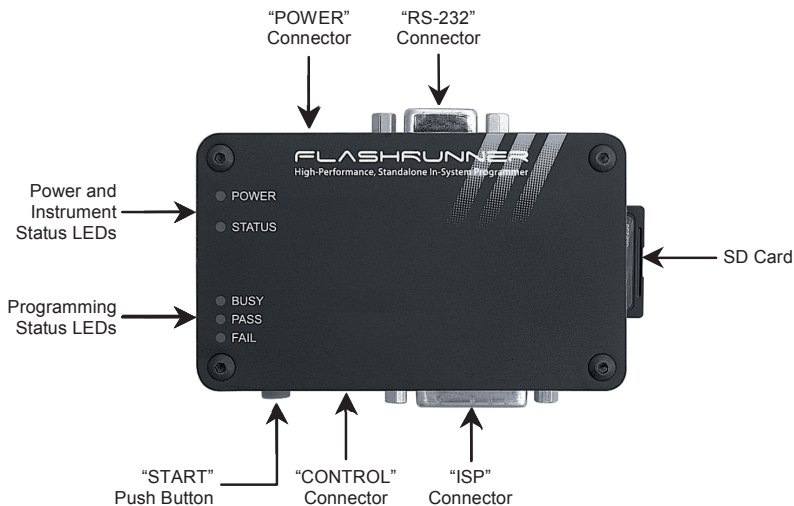
Figure 1.2: Hardware Overview

1

1.4 Programming Algorithms and Licenses

FlashRunner FR03 comes already preinstalled with programming algorithms
(and relative programming licenses) for the silicon manufacturer you
specified at the moment of purchase. While you can't install new licenses,
the programming algorithms can be updated if new versions become
available.

Programming algorithms and license files are stored in the SD card (see the
FlashRunner Programmer's Manual for more information).

> **i** **Note:** *FlashRunner FR03 uses a proprietary file system. The
> contents of the FlashRunner FR03 SD card are not readable
> by a PC.*

1.5 Upgrading the Firmware

The FlashRunner firmware can be easily upgraded using the provided
Control Panel utility. For more information, please refer to the FlashRunner
Programmer's Manual.

2 System Setup

2.1 Overview

> **i** **Note:** *the example shows how to set up the system for programming a Freescale MC68HC908QY4 microcontroller. For how to connect to other target devices, please refer to the FlashRunner Programmer's Manual.*

2

This chapter will explain how to set up FlashRunner FR03 for the first time. Although FlashRunner is typically used for standalone operations (Standalone mode), the examples in this chapter will use the host system to send commands to FlashRunner (Host mode).

When moving FlashRunner to the production environment, you can take full advantage of the instrument's SD card to make the instrument work without being controlled by the host system.

For more information about Standalone mode and Host mode, see the FlashRunner Programmer's Manual.

2.2 Software Setup

The FlashRunner system software setup installs all of the required components to your hard drive. These components include:

- The FlashRunner Control Panel utility;
- Script examples;
- Documentation in PDF format.

To install the FlashRunner system software:

- Insert the **"System Software"** CD-ROM into your computer's CD-ROM drive;
- A startup window will automatically appear. Choose **"Install Instrument Software"** from the main menu. Follow the on-screen instructions.

2

> **i** **Note:** *to install the FlashRunner system software on Windows 2000 or Windows XP, you must log in as Administrator.*

2.3 Hardware Setup

To set up FlashRunner FR03, you must follow the steps below, in the indicated order:

1. Interface FlashRunner with your test/programming equipment;
2. Connect FlashRunner to the host system;
3. Power up FlashRunner;
4. Send FlashRunner commands via the FlashRunner Control Panel utility.

2.3.1 Interfacing FlashRunner with Your Test/Programming Equipment

FlashRunner FR03 connects to your target board through signals grouped in the "ISP" connector.

The specific ISP signals that must be routed from FlashRunner FR03 to your target board depend on the specific target device. Typical connections for all the device families supported by FlashRunner are shown in the FlashRunner Programmer's Manual.

Additional control signals (START, BUSY, PASS, FAIL) are grouped in the "CONTROL" connector, and can be used if you want your test/programming equipment to control FlashRunner through these lines.

2.3.2 Connecting FlashRunner to the Host System

FlashRunner FR03 connects to the host system via RS-232.

2.3.3 Powering Up FlashRunner

Power up FlashRunner by connecting the output of a 7.5V power supply (not included) to the "POWER" barrel connector. FlashRunner accepts any DC voltage between 6.5V and 9V.

2

2.4 Step-by-Step Tutorial: Sending Commands to FlashRunner

After setting up the hardware, you are ready to send commands to the instrument. The following steps will guide you through the process of launching your first FlashRunner commands using the provided FlashRunner Control Panel utility. For detailed information about the FlashRunner Control Panel utility, see the FlashRunner Programmer's Manual.

> **i**
>
> **Note:** *the following steps show how to program a Freescale MC68HC908QY4 microcontroller, and the details are therefore specific for that microcontroller. However, the procedures shown are general and will allow you get a feel of how FlashRunner works.*

1. Launch the FlashRunner Control Panel utility. Select **Start > Programs > SMH Technologies > F lashRunner > Control Panel**. The Control Panel utility will open.

    To establish a connection with FlashRunner, on the **"Communication Settings"** section, select **"FlashRunner serial version"**, and specify the COM port you are using and the baud rate (by default, FlashRunner communicates at 115200 bps).
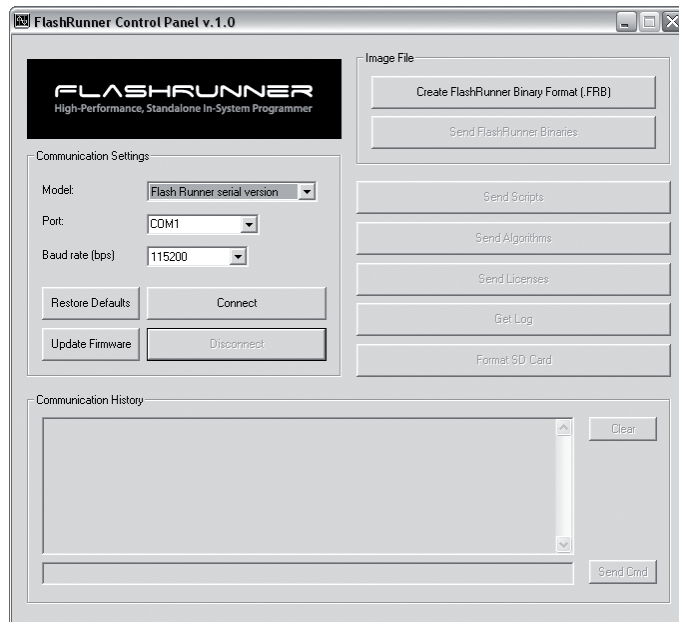
Figure 2.1: FlashRunner Control Panel, Communication Settings

2.   Click the **"Connect"** button. On the **"Communication History"**
     section, note the commands that have been sent and received. In this
     case, the **SPING** command is automatically sent to FlashRunner,
     which replies with the **PONG>** string.

3.   In the edit box below the communication history, type the following
     commands (each followed by Return):

```
TCSETDEV FREESCALE MC68HC908QY4 HC08
TCSETPAR FOSC 16000000
TCSETPAR FDIV 4
TCSETPAR VDD 5000
```

These commands set, respectively, the target microcontroller, the
oscillator frequency, the internal divisor and the VDD voltage. In this
example, we used a 16 MHz oscillator, the internal divisor for
MC68HC908QY4 devices is fixed to 4, and the VDD is 5 V.

FlashRunner will respond to each command with the **>** string, indicating that the command has been successfully executed. After sending these commands, the Control Panel will look like the figure below.
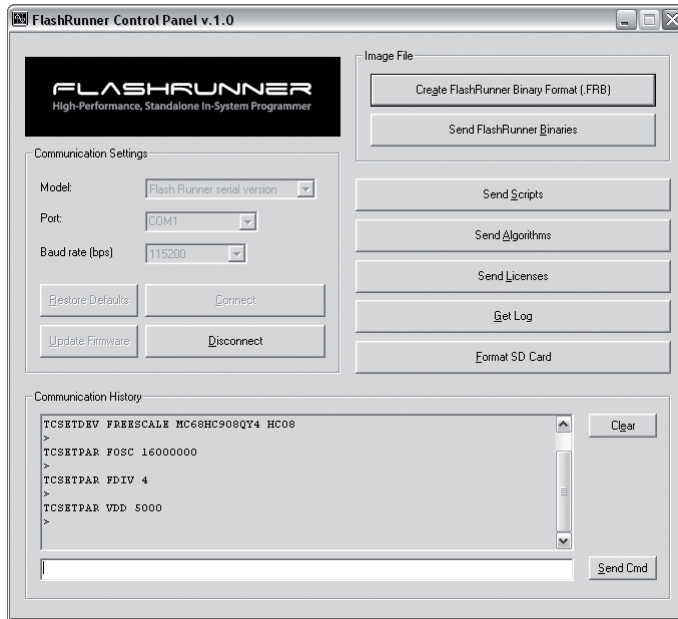
Figure 2.2: FlashRunner Control Panel, Target Device Configured

4. When working with Freescale HC08 devices, FlashRunner requires you to specify the Reset power up and power down times, in milliseconds. Send the following two commands:

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

5. After specifying up the target device settings, we are ready to transfer to FlashRunner the binary image to be programmed into the target device. FlashRunner accepts only image files in a .frb (FlashRunner Binary) format. To convert your binary, Intel-Hex or S19 image file to the

FlashRunner format, click the **"Create FlashRunner Binary Format"** button. The following dialog box will appear.

2
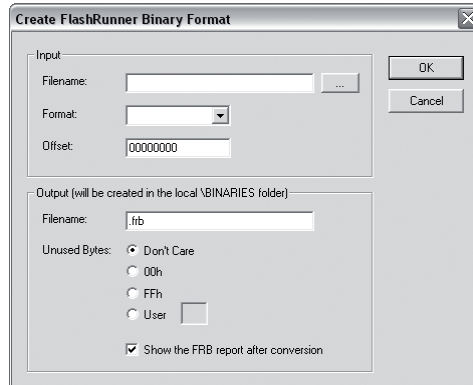
Figure 2.3: FlashRunner Control Panel, Binary File Conversion

In the **"Input"** section, specify the source file to be converted, its format, and the address from which the file conversion will start (offset). In the **"Output"** section, specify the output filename and the value used to fill unused location.

Click the **"OK"** button. The FlashRunner Binary file will be created in the local **\BINARIES** folder.

6.   To transfer the created image to FlashRunner, send the following command:

```
TPSENDFILE YMODEM DEMO.FRB
```

In this example, the image file is called **DEMO.FRB**. The following dialog box will appear.
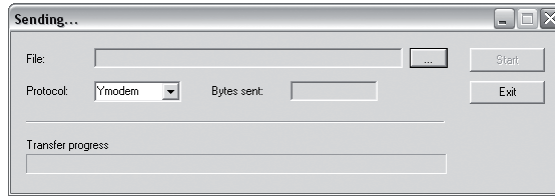
Figure 2.4: FlashRunner Control Panel, File Transfer

2

Click the **"..."** to browse for the image file to be set, then click **"Start"** to begin the transfer. The file will be saved to the FlashRunner SD card, in the `\BINARIES` folder.

7. We are now ready to start the actual programming part. Send the following commands:

```
TPSETSRC FILE DEMO.FRB
TPSTART
TPCMD SETPWD CONST $FF $FF $FF $FF $FF $FF $FF $FF
TPCMD MASSERASE F
TPCMD BLANKCHECK F $EE00 4608
TPCMD PROGRAM F $EE00 $EE00 4608
TPCMD VERIFY F S $EE00 $EE00 4608
TPEND
```

The data to be programmed is taken from the image file starting at $EE00 (offset from the beginning of the file), is programmed to the target microcontroller starting from the location $EE00 and is 4608 bytes long.

The `TPSETSRC` command specifies the source file for the `TPCMD PROGRAM` and `TPCMD VERIFY` commands that come next. All the actual programming operations are sent between a `TPSTART` and `TPEND` command. The `TPCMD SETPWD` command sets the security bytes needed to perform subsequent operations.

After sending these commands, the Control Panel will look like the figure below.
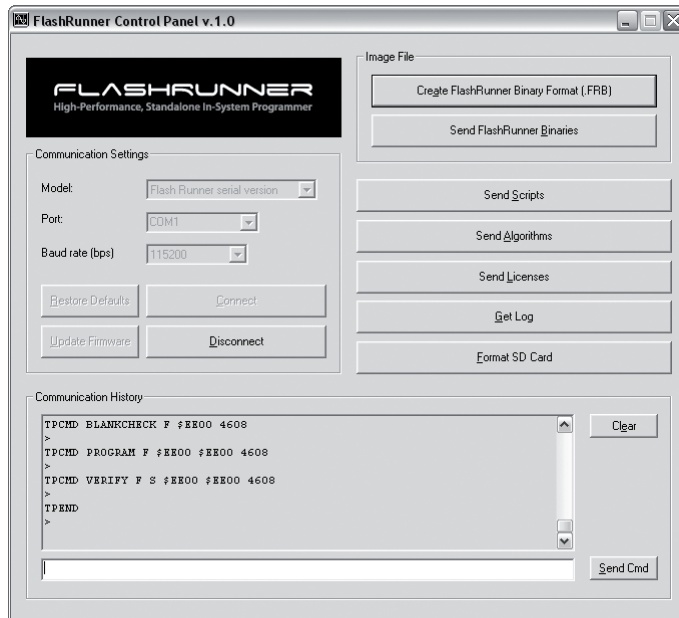
Figure 2.5: FlashRunner Control Panel, Target Device Programmed

8. We are now done with programming the target device. Click the **"Disconnect"** button to free the serial port resource.

For detailed information on all of the FlashRunner commands and their syntax, including specific commands for specific family of microcontrollers, please refer to the FlashRunner Programmer's Manual, included (in PDF format) in the FlashRunner CD-ROM.

Programming can be automated by creating "scripts". Scripts are text files, stored in the SD card, which contain a sequence of FlashRunner commands. See the FlashRunner Programmer's Manual for more information about scripts.

3 Connectors

3.1 Overview

FlashRunner FR03 connects to your programming/testing system through two connectors: the "ISP" D-Sub connector groups ISP signals, and the "CONTROL" header connector groups control signals.

The "RS-232" connector allows FlashRunner to be set up and controlled by a host system.

FlashRunner is powered through the "POWER" connector.

3

> ⚠ **Warning:** *the "ISP", "CONTROL", "RS-232" and "POWER" connectors share the same ground reference. Care must be taken when a host PC or ATE equipment with a ground reference different from the target board's is connected to the "CONTROL" and/or "RS-232" connectors.*

3.2 ISP Connector

The "ISP" connector groups the signals needed to program the target device. This connector groups several input/output lines, both digital and analog, that are automatically configured by FlashRunner depending on the specific target device to be programmed (see the FlashRunner Programmer's Manual to learn how to connect these lines to your specific target device).

> ⚠ **Warning:** *in order to avoid undesired current loops between the FlashRunner power supply and the target board, a power supply with a floating output (ground not referenced to the earth potential) should be used.*

3

> ⚠ **Note:** *when FlashRunner is powered off, ISP signals are not HiZ. As a result, ISP signals have a low impedance that could be intrusive when other tools perform in-circuit testing on the target board.*
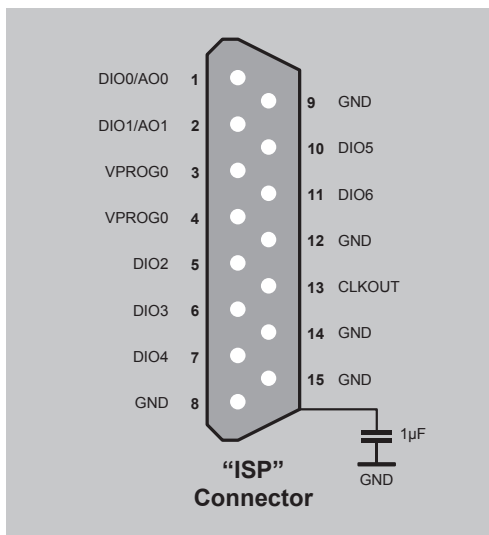
Figure 3.1: ISP Connector

Table 3.1: ISP Connector Signals

Pin #	Signal Name	Description
1	DIO0/AO0	Digital input/output 0 or analog output 0
2	DIO1/AO1	Digital input/output 1 or analog output 1
3	VPROG0	Programmable voltage 0 (max 5.5V, 500mA)
4	VPROG0	Programmable voltage 0 (max 5.5V, 500mA)
5	DIO2	Digital input/output 2
6	DIO3	Digital input/output 3
7	DIO4	Digital input/output 4
8	GND	Ground
9	GND	Ground
10	DIO5	Digital input/output 5
11	DIO6	Digital input/output 6
12	GND	Ground
13	CLKOUT	Clock output
14	GND	Ground
15	GND	Ground

3

3.3 Control Connector

The "CONTROL" connector is used by FlashRunner for integration with an automatic programming/testing equipment.
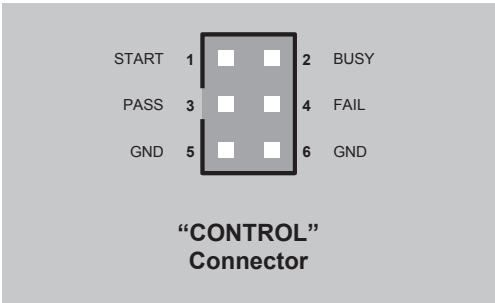
Figure 3.2: Control Connector

Table 3.2: Control Connector Signals

Pin #	Signal Name	Description
1	START	START (input , active low)
2	BUSY	BUSY (output, push-pull, active low)
3	PASS	PASS (output, push-pull, active low)
4	FAIL	FAIL (output, push-pull, active low)
5	GND	Ground
6	GND	Ground

3

3.4 RS-232 Connector

The "RS-232" D-Sub connector can be used to communicate with a host system.
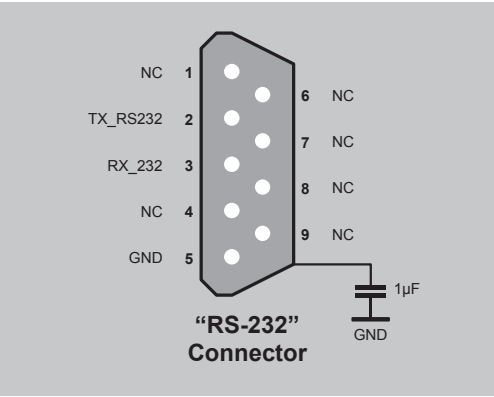
Figure 3.3: RS-232 Connector

Table 3.3: RS-232 Connector Signals

Pin #	Signal Name	Description
1	NC	Not connected
2	TX_RS232	TX (output, RS-232 levels)
3	RX_RS232	RX (input, RS-232 levels)
4	NC	Not connected
5	GND	Ground
6	NC	Not connected
7	NC	Not connected
8	NC	Not connected
9	NC	Not connected

3

3.5 Power Connector

FlashRunner is powered by an external power supply through the "POWER" connector. The "POWER" connector (2.1 mm pin and sleeve plug with positive in the center and sleeve as ground) accepts any DC voltage between 6.5V and 9V.

GND — 6.5V - 9V DC
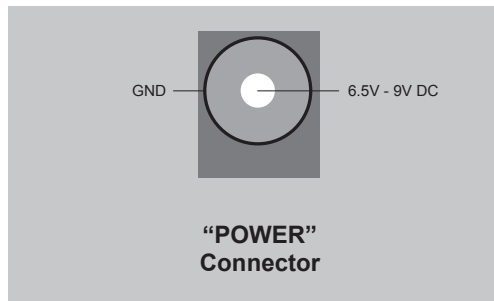
**"POWER"
Connector**

Figure 3.4: Power Connector

4 Technical Specifications

4.1 Absolute Maximum Ratings

Table 4.1: Absolute Maximum Ratings

Parameter	Value
"CONTROL" Connector	
Maximum input voltage on line START	-0.3V to +6V
Maximum current on lines BUSY, PASS,FAIL	-25mA to 25mA
"ISP" Connector	
Maximum input voltage on lines DIO/AO[1..0], DIO[6..2], CLKOUT	-1V to +7V
Maximum current on lines DIO/AO[1..0], DIO[6..2], CLKOUT	±50mA
Maximum current on line VPROG0	500mA
"RS-232" Connector	
Maximum input voltage on line RX_RS232	-25V to +25V
Maximum current on line TX_RS232	±60mA
"POWER" Connector	
Maximum supply voltage	-20V to +12V

4

4.2 DC Characteristics and Functional Operating Range

Table 4.2: DC Characteristics and Functional Operating Range

Parameter	Condition	Value		
		Min	Typ	Max
"CONTROL" Connector				
V_{IL} (input low voltage) on line START		-0.3V	-	1.75V
V_{IH} (input high voltage) on line START		3.25V	-	5.3V
V_{OL} (output low voltage) on lines BUSY, FAIL, PASS	I_{OL} = 10mA	0V	-	0.8V
V_{OH} (output high voltage) on lines BUSY, FAIL, PASS	I_{OH} = 10mA	4.2V	-	5V
"ISP" Connector				
V_{IL} (input low voltage) on lines DIO[6..2], DIO[1..0]	Configured as digital lines	-	-	$0.3V_{PROG0}$
V_{IH} (input high voltage) on lines DIO[6..2], DIO[1..0]	Configured as digital lines	$0.7V_{PROG0}$	-	V_{PROG0}
V_{OL} (output low voltage) on lines DIO[6..2], DIO[1..0], CLKOUT	Configured as digital lines, V_{PROG0} = 3V, I_{OL} = 12mA	-	-	0.36V
V_{OH} (output high voltage) on lines DIO[6..2], DIO[1..0], CLKOUT	Configured as digital lines, V_{PROG0} = 3V, I_{OH} = 12mA	2.56V	-	-
V_{OL} (output low voltage) on lines DIO[6..2], DIO[1..0], CLKOUT	Configured as digital lines, V_{PROG0} = 5.5V, I_{OL} = 24mA	-	-	0.36V
V_{OH} (output high voltage) on lines DIO[6..2], DIO[1..0], CLKOUT	Configured as digital lines, V_{PROG0} = 5.5V, I_{OH} = 24mA	4.86V	-	-
I_{OH} current (source) on lines DIO[6..2], DIO[1..0]	Configured as input with active pull-ups	-	3.4mA	-
DIO/AO[1..0] voltage	Configured as analog output	3V	-	14.5V
DIO/AO[1..0] IO current (sink and source)	Configured as analog output	-	-	±40mA
I_{OH} current (source) on lines DIO/AO[1..0]	Configured as analog lines with active pull-ups	-	5.5mA	-
VPROG0 line output voltage		1.6V	-	5.5V
VPROG0 current (source)		-	-	500mA
"RS-232" Connector				
V_{IL} (input low voltage) on line RX_RS232		-	-	1.2V
V_{IH} (input high voltage) on line RX_RS232		2.4V	-	-
V_{OL} (output low voltage) on line TX_RS232	R_{LOAD} = 3KΩ	-	-	-5V
V_{OH} (output high voltage) on line TX_RS232	R_{LOAD} = 3KΩ	+5V	-	-
"POWER" Connector				
POWER line input voltage		6.5V	7.5V	9V
POWER line consumption		-	-	1.5A

4

4.3 AC Characteristics

The following table summarizes AC characteristics for ISP lines on the "ISP" connector.

Table 4.3: AC Characteristics

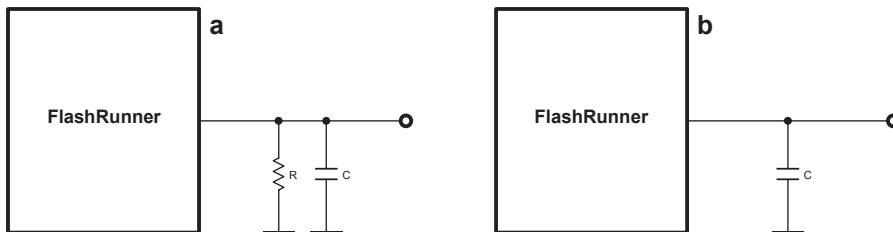| Parameter	Condition		Value		
			Min	Typ	Max
t_{RISE} on lines DIO[6..2], DIO[1..0], CLKOUT when configured as digital output push-pull	V_{PROG0} = 1.8V	Load: 470Ω//100pF (see figure 4.1a)	-	40ns	-
	V_{PROG0} = 3.3V		-	30ns	-
	V_{PROG0} = 5V		-	25ns	-
t_{FALL} on lines DIO[6..2], DIO[1..0], CLKOUT when configured as digital output push-pull	V_{PROG0} = 1.8V	Load: 470Ω//100pF (see figure 4.1a)	-	35ns	-
	V_{PROG0} = 3.3V		-	25ns	-
	V_{PROG0} = 5V		-	25ns	-
t_{RISE} on lines DIO/AO[1..0] configured as analog output	V_{OUT} = 3V	Load: 4.7KΩ//100pF (see figure 4.1a)	-	7µs	-
	V_{OUT} = 12V		-	11µs	-
	V_{OUT} = 14.5V		-	12µs	-
t_{FALL} on lines DIO/AO[1..0] configured as analog output	V_{OUT} = 3V	Load: 100pF (see figure 4.1b)	-	8µs	-
	V_{OUT} = 12V		-	20µs	-
	V_{OUT} = 14.5V		-	30µs	-
t_{RISE} on line VPROG0	V_{PROG0} = 0-1.8V	Load: 15Ω//10mF (see figure 4.1a)	-	10ms	-
	V_{PROG0} = 0-3.3V	Load: 22Ω//10mF (see figure 4.1a)	-	15ms	-
	V_{PROG0} = 0-5.5V	Load: 22Ω//10mF (see figure 4.1a)	-	20ms	-
t_{FALL} on line VPROG0	V_{PROG0} = 1.8-0V	Load: 10mF (see figure 4.1b)	-	300ms	-
	V_{PROG0} = 3.3-0V		-	350ms	-
	V_{PROG0} = 5.5-0V		-	350ms	-
CLKOUT frequency			0MHz	-	50MHz

4

Figure 4.1: Load Conditions

4.4 Physical and Environmental Specifications

Table 4.5: Physical and Environmental Specifications

Parameter	Value
Dimensions	89 x 65 x 35 mm
Weight	135 g
"ISP" connector type	15-pin, D-Sub (female)
"CONTROL" connector type	6-pin, 2.54mm-pitch, dual-row boxed header (male)
"RS-232" connector type	9-pin, D-Sub (female)
"POWER" connector type	Barrel connector, 2.1mm central pin
Operating temperature	0-50°C
Operating humidity	90% max (without condensation)
Storage temperature	0-70°C
Storage humidity	90% max (without condensation)

4