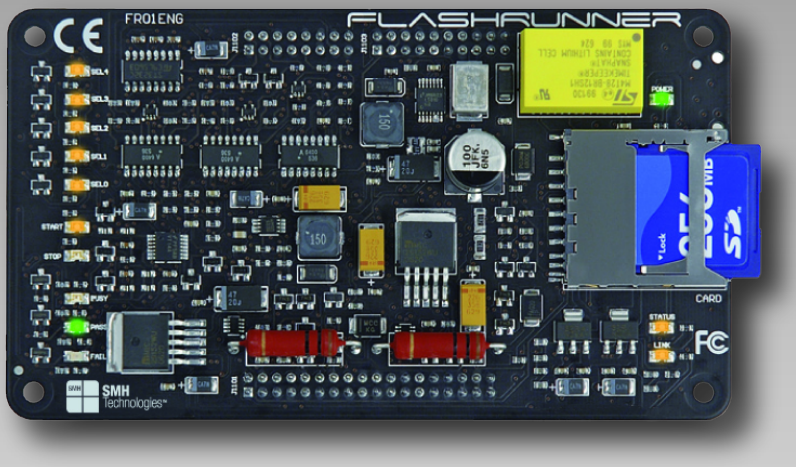
FLASHRUNNER

High-Performance, Standalone In-System Programmer

# FR01ENG

# User's Manual




SMH Technologies™

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

# FlashRunner FR01ENG

## High-Performance, Standalone In-System Programmer

## User's Manual

Revision 1.3 - April 2015

SMH Technologies™

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING


DC10732

***We want your feedback!***

*SMH Technologies is always on the lookout for new ways to improve its Products and Services. For this reason feedback, comments, suggestions or criticisms, however small, are always welcome.*

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

# Index of Figures

# Index of Tables

# 0 Before Starting

**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 ground (e.g. via wrist straps) when handling the instrument.

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

## 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 FR01ENG 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 What is FlashRunner FR01ENG?

FlashRunner FR01ENG is a member of the FlashRunner series of a high-performance, standalone In-System Programmers specific for Flash-based microcontrollers and serial memories.

FlashRunner FR01ENG is targeted at production environments and piggybacks to your programming system or Automatic Test Equipment (ATE) via header connectors, and can work either in full standalone mode or controlled by a host system.

### 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

- 9 to 24V power supply input;
- Five digital I/O lines;
- Two digital I/O or analog output lines;
- Two programmable output voltages (0 to 15V, 0.25A and 0 to 5V, 0.5A);

- One analog input line;
- One programmable clock output;
- Secure Digital memory card (up to 2 GB);
- 512 bytes on-board dynamic memory;
- On-board timekeeper and calendar;
- I/O protection;
- Optoisolated inputs for project selection;
- Two optoisolated command inputs (START and STOP);
- Three optoisolated status outputs (FAIL, PASS, BUSY);
- One optoisolated RS-232 channel.

### 1.1.3 Software features

- Fully autonomous standalone mode thanks to its SD memory card (FAT16);
- Controllable by any host system through a terminal utility and simple ASCII protocol;
- Up to 32 hardware-selectable projects (scripts), unlimited software-selectable projects;
- Log files;
- Erase, blank check, program, read, verify, oscillator trimming, etc.

## 1.2 Package Checklist

The FlashRunner FR01ENG package includes the following items:

- FlashRunner FR01ENG unit, including an SD card already pre-installed with the programming algorithm(s) you specified at the time of purchase;
- FlashRunner “System Software” CD-ROM, containing the FlashRunner Control Panel utility and the FlashRunner Programmer’s Manual in PDF format;
- This user’s manual;
- A registration card.

FlashRunner FR01ENG is composed of two layers: a bottom layer and a top layer.

The bottom layer contains all of the FlashRunner electronics; the top layer has the function of protecting the bottom layer and replicating the status LEDs.

If space is an issue when integrating FlashRunner in your programming/testing system, the top layer can be easily removed.

The following figures show FlashRunner's top and bottom layers.


Figure 1.1: FlashRunner Top Layer

1

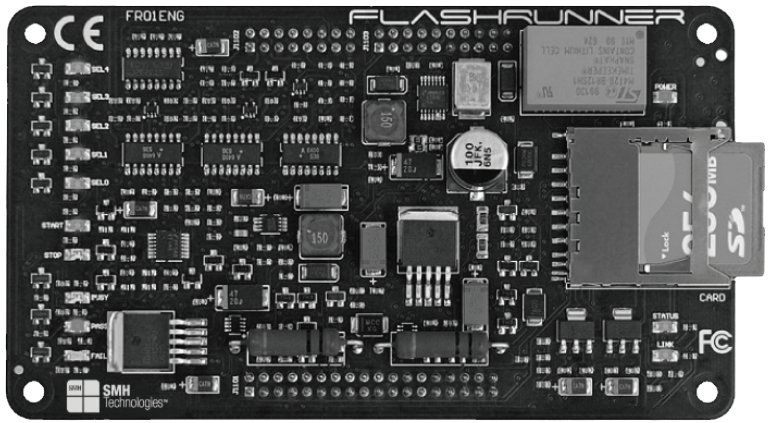



Figure 1.2: FlashRunner Bottom Layer (LEDs Side)

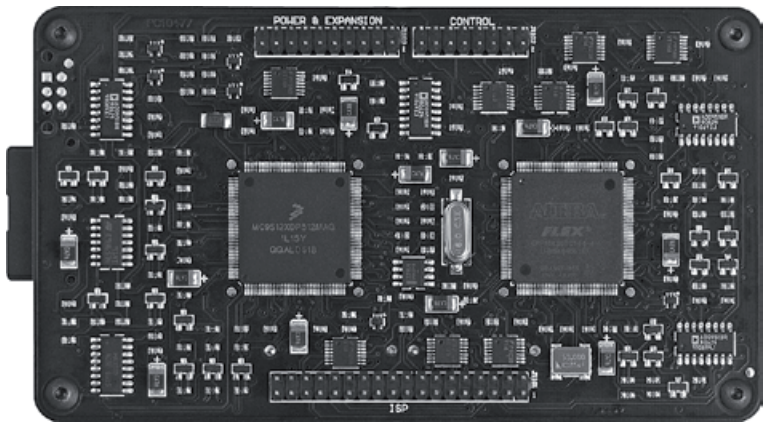



Figure 1.3: FlashRunner Bottom Layer (Connectors Side)

## 1.3 Programming Algorithms and Licenses

FlashRunner FR01ENG includes programming algorithms for several devices. In order to program a specific device, however, a specific license file for that device must be purchased.

**Note:** *FlashRunner FR01ENG comes already preinstalled with the license(s) you specified at the moment of purchase. You can purchase additional licenses at any future moment.*



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

### 1.3.1 Installing New Licenses

When you buy an additional license for a specific device, you will get:

- An algorithm file (.alg);
- A license file (.lic);
- A device-specific script example (.frs).

The .alg file contains the actual programming algorithm for the requested device (and several other devices of the same family).

The .lic file contains an unlocking code that will let you use the programming algorithm. A license file enables the use of a specific programming algorithm on a specific FlashRunner instrument (licenses are serial number specific).

The script file contains an example of script to use as a starting point for your specific programming needs (for more information on scripts, see the FlashRunner Programmer's Manual).

To install the new license, do the following:

1. Copy the **`.alg`** file into the **`\ALGOS`** directory of the SD card (if an .alg file with the same name already exists, overwrite it);
2. Copy the **`.lic`** file into the **`\LICENSES`** directory of the SD card.

To copy files on the SD card, use either a standard card reader connected to a PC or transfer the files using the FlashRunner **`FSSENDFILE`** command (for more information on FlashRunner commands, see the FlashRunner Programmer's Manual).

Alternatively, you can use the FlashRunner Control Panel utility to install new programming algorithms and licenses. For more information on the FlashRunner Control Panel please refer to the FlashRunner Programmer's Manual.

## 1.4 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



**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.*

This chapter will explain how to set up FlashRunner FR01ENG 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.



**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 FR01ENG, you must follow the steps below, in the indicated order:

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

### 2.3.1 Interfacing FlashRunner with your Test/Programming Equipment

FlashRunner FR01ENG typically piggybacks to a carrier board (designed by you) that interfaces FlashRunner to your test/programming equipment and to your target board. The carrier board must include three female header connectors that accept FlashRunner's "ISP", "CONTROL" and "POWER" connectors. The carrier board:

- Supplies power to FlashRunner. FlashRunner accepts any DC voltage between 9 V and 24 V;
- Routes all of the required ISP and power signals from FlashRunner to the target board;
- Routes the RS-232 signals from FlashRunner "CONTROL" connector to a RS-232 connector (for communication with a host PC).

- Routes control signals (START, STOP, BUSY, PASS, FAIL and script selection lines) to your test/programming equipment (only necessary for standalone operations).

Connections between the carrier board and the target board typically consist of flat cables that plug into the ISP connector on the target board(s) or single wires that connect to the single nails of your bed-of-nails fixture.

The figure below illustrates typical programming connections.



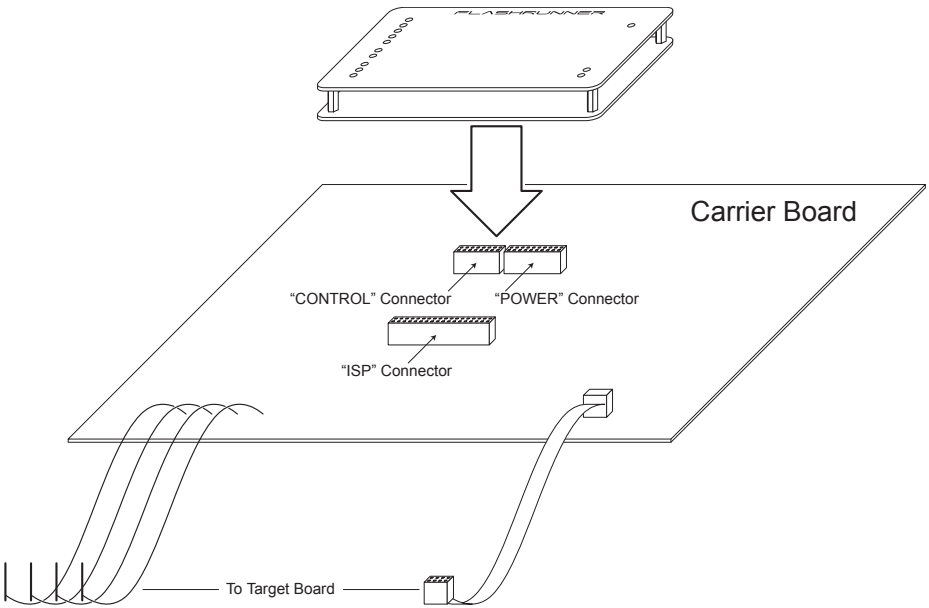



Figure 2.1: Typical Programming Connections

The specific ISP signals that must be routed from FlashRunner FR01ENG 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.

### 2.3.2 Connecting FlashRunner to the Host PC System

To connect FlashRunner FR01ENG to a host PC, you must provide a RS-232 connector in your carrier board first, routing the appropriate signals from FlashRunner's "CONTROL" connector to the RS-232 connector.

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.

**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 > FlashRunner > Control Panel**. The Control Panel utility will open.
2. 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).

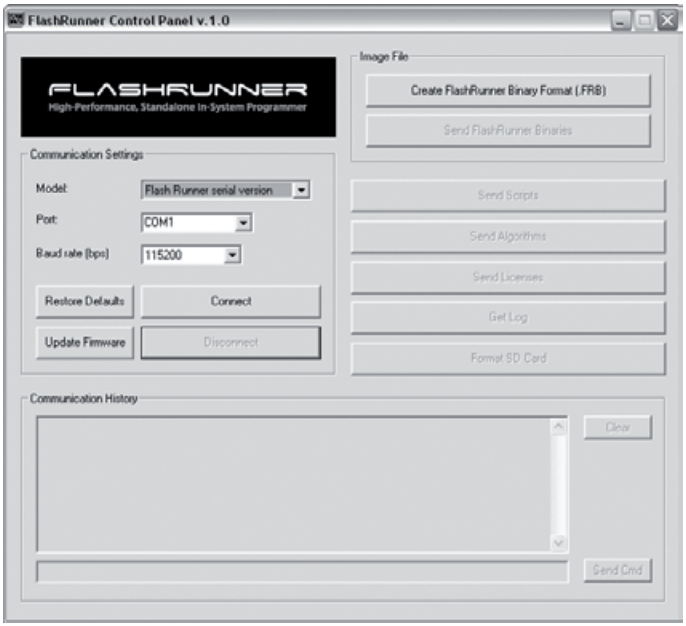



Figure 2.2: FlashRunner Control Panel, Communication Settings

3. 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.
4. 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.

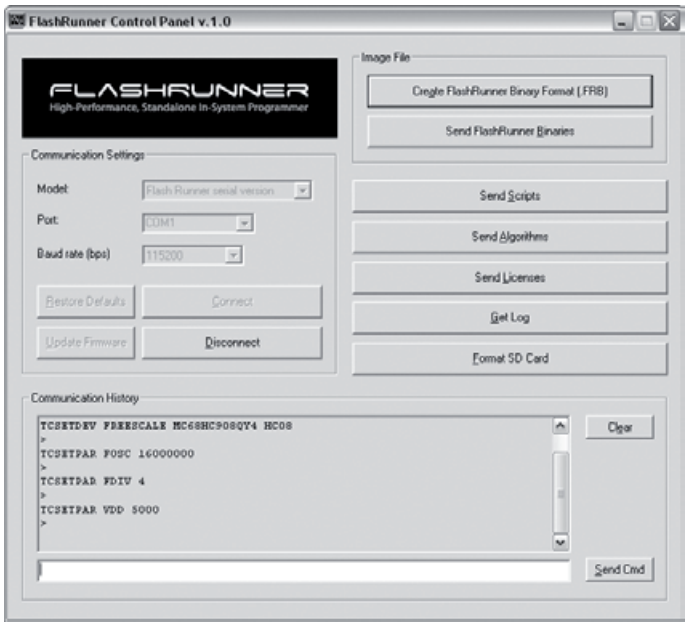



Figure 2.3: FlashRunner Control Panel, Target Device Configured

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

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

6. After specifying 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.

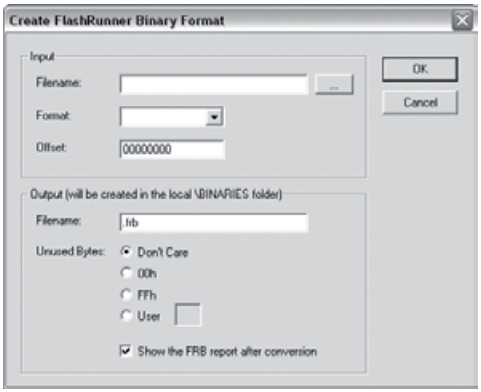



Figure 2.4: FlashRunner Control Panel, Binary File Conversion

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 locations.

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

7. 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.

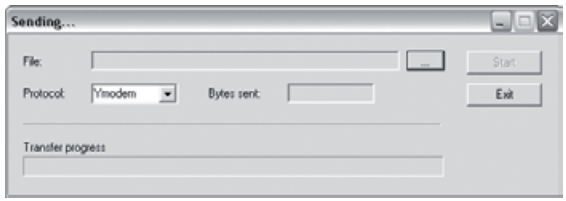



Figure 2.5: FlashRunner Control Panel, File Transfer



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

8. 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`** e **`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.

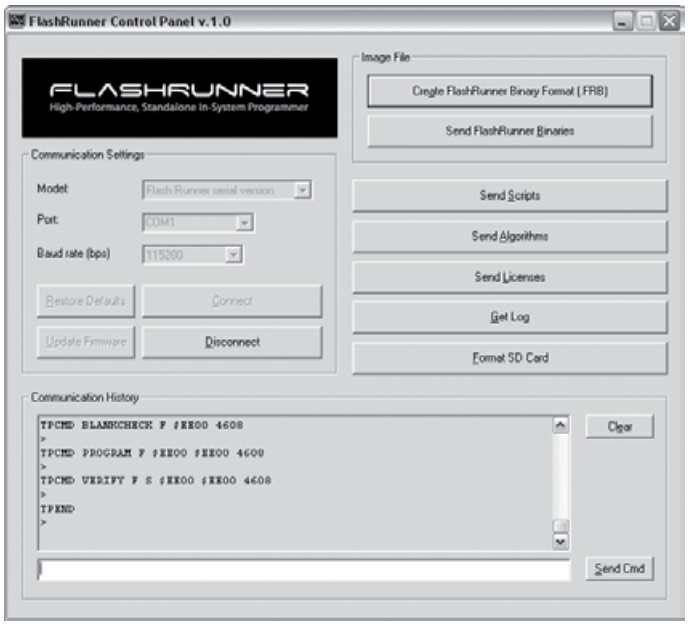



Figure 2.6: FlashRunner Control Panel, Target Device Programmed

9. 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 connects to your programming/testing system through three header connectors: one groups ISP signals, one groups control signals, and one groups power signals.



## 3.2 Power Connector

The “POWER” connector is used to power FlashRunner. This connector also includes reserved expansion lines, which must not be connected.

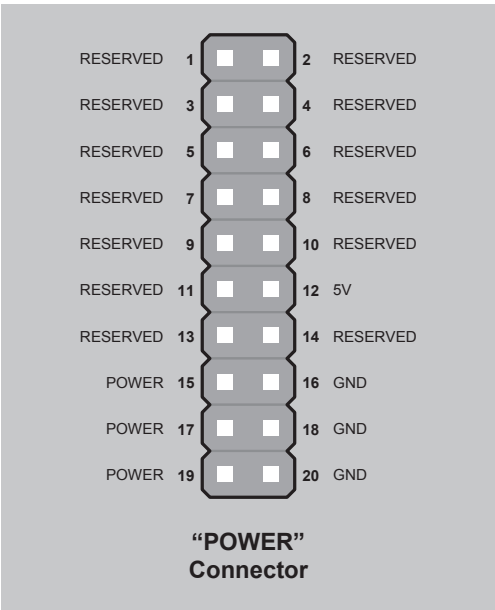



Figure 3.1: Power Connector

Table 3.1: Power Connector Signals

| Pin # | Signal Name | Description |
|---|---|---|
| 1 | RESERVED | Internal line for future expansion. Do not connect. |
| 2 | RESERVED | Internal line for future expansion. Do not connect. |
| 3 | RESERVED | Internal line for future expansion. Do not connect. |
| 4 | RESERVED | Internal line for future expansion. Do not connect. |
| 5 | RESERVED | Internal line for future expansion. Do not connect. |
| 6 | RESERVED | Internal line for future expansion. Do not connect. |
| 7 | RESERVED | Internal line for future expansion. Do not connect. |
| 8 | RESERVED | Internal line for future expansion. Do not connect. |
| 9 | RESERVED | Internal line for future expansion. Do not connect. |
| 10 | RESERVED | Internal line for future expansion. Do not connect. |
| 11 | RESERVED | Internal line for future expansion. Do not connect. |
| 12 | 5V | User power supply (output, 5V) |
| 13 | RESERVED | Internal line for future expansion. Do not connect. |
| 14 | RESERVED | Internal line for future expansion. Do not connect. |
| 15 | POWER | FlashRunner power supply (input, 9-24V) |
| 16 | GND | Ground |
| 17 | POWER | FlashRunner power supply (input, 9-24V) |
| 18 | GND | Ground |
| 19 | POWER | FlashRunner power supply (input, 9-24V) |
| 20 | GND | Ground |

## 3.3 Control Connector

The “CONTROL” connector is used by FlashRunner to communicate with the host system and for integration with an automatic programming/testing equipment.

**Note:** *all control signals are optoisolated. You must power the optoisolation circuitry through the OPTO_5V and OPTO_GND lines.*

*If your system doesn’t require optoisolation, just connect the OPTO_5V and OPTO_GND lines to the 5V and GND lines (respectively) of the “POWER” connector.*

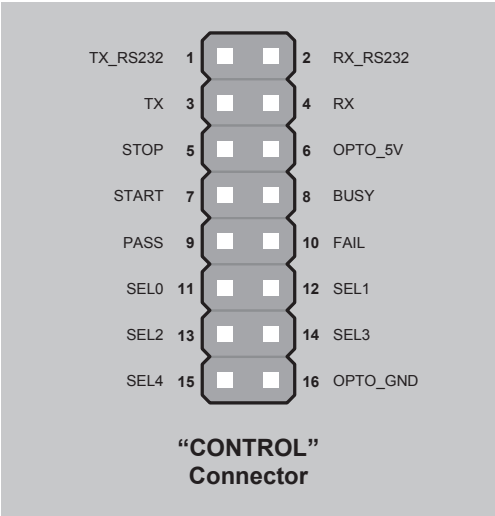



Figure 3.2: Control Connector

Table 3.2: Control Connector Signals

| Pin # | Signal Name | Description |
|---|---|---|
| 1 | TX_RS232 | TX (output, optoisolated, RS-232 levels) |
| 2 | RX_RS232 | RX (input, optoisolated, RS-232 levels) |
| 3 | TX | TX (output , open-drain, optoisolated, 0-5V levels) |
| 4 | RX | RX (input, optoisolated, 0-5V levels) |
| 5 | STOP | STOP (input , optoisolated, active low) |
| 6 | OPTO_5V | Optoisolation power supply (input, 5V). If your system doesn't require optoisolation, connect this line to the "5V" lines of the "POWER" connector. |
| 7 | START | START (input , optoisolated, active low) |
| 8 | BUSY | BUSY (output, open-drain, optoisolated, active low) |
| 9 | PASS | PASS (output, open-drain, optoisolated, active low) |
| 10 | FAIL | FAIL (output, open-drain, optoisolated, active low) |
| 11 | SEL0 | Script selection 0 (input, optoisolated) |
| 12 | SEL1 | Script selection 1 (input, optoisolated) |
| 13 | SEL2 | Script selection 2 (input, optoisolated) |
| 14 | SEL3 | Script selection 3 (input, optoisolated) |
| 15 | SEL4 | Script selection 4 (input, optoisolated) |
| 16 | OPTO_GND | Optoisolation ground. If your system doesn't require optoisolation, connect this line to the "GND" lines of the "POWER" connector. |



## 3.4 ISP Connector

The "ISP" connector has all of the signals needed to program the target device. This connector has several input/output lines, both digital and analog, which 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).

**Note:** *ISP signals are not optoisolated.*

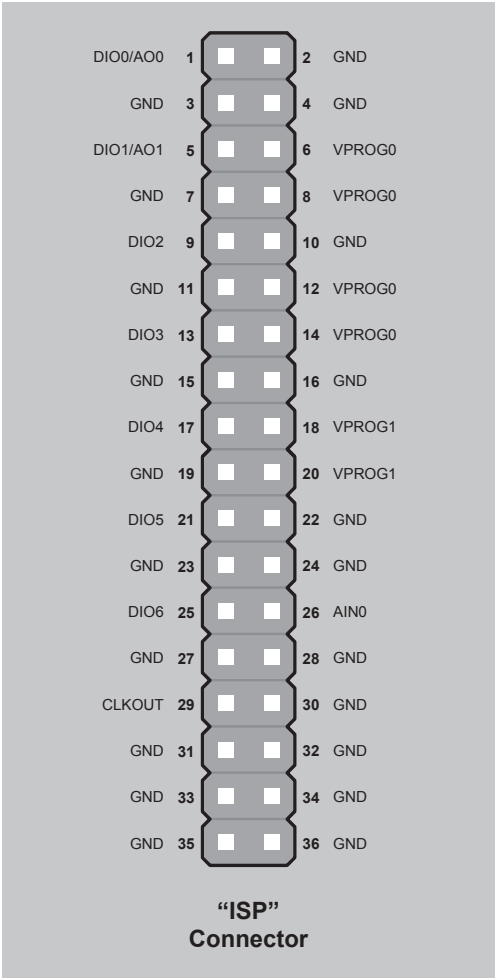



Figure 3.3: ISP Connector

Table 3.3: ISP Connector Signals

| Pin # | Signal Name | Description |
|---|---|---|
| 1 | DIO0/AO0 | Digital input/output 0 or analog output 0 |
| 2 | GND | Ground |
| 3 | GND | Ground |
| 4 | GND | Ground |
| 5 | DIO1/AO1 | Digital input/output 1 or analog output 1 |
| 6 | VPROG0 | Programmable voltage 0 (max 5.5V, 500mA) |
| 7 | GND | Ground |
| 8 | VPROG0 | Programmable voltage 0 (max 5.5V, 500mA) |
| 9 | DIO2 | Digital input/output 2 |
| 10 | GND | Ground |
| 11 | GND | Ground |
| 12 | VPROG0 | Programmable voltage 0 (max 5.5V, 500mA) |
| 13 | DIO3 | Digital input/output 3 |
| 14 | VPROG0 | Programmable voltage 0 (max 5.5V, 500mA) |
| 15 | GND | Ground |
| 16 | GND | Ground |
| 17 | DIO4 | Digital input/output 4 |
| 18 | VPROG1 | Programmable voltage 1 (max 14.5V, 250mA) |
| 19 | GND | Ground |
| 20 | VPROG1 | Programmable voltage 1 (max 14.5V, 250mA) |
| 21 | DIO5 | Digital input/output 5 |
| 22 | GND | Ground |
| 23 | GND | Ground |
| 24 | GND | Ground |
| 25 | DIO6 | Digital input/output 6 |
| 26 | AIN0 | Analog input 0 (max 28.5V) |
| 27 | GND | Ground |
| 28 | GND | Ground |
| 29 | CLKOUT | Clock output |
| 30 | GND | Ground |
| 31 | GND | Ground |
| 32 | GND | Ground |
| 33 | GND | Ground |
| 34 | GND | Ground |
| 35 | GND | Ground |
| 36 | GND | Ground |

## 3.5 ATE Connection Example

The figure below shows an example of connection between FlashRunner and an ATE system. In this example, the target board is automatically powered by FlashRunner through the VPROG0 line.

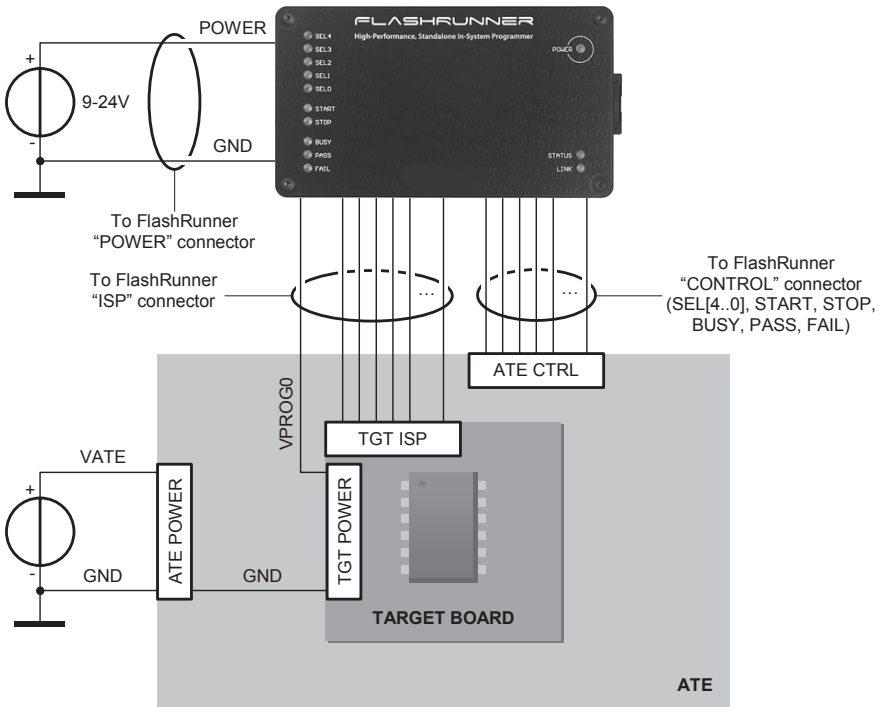



Figure 3.4: Example of Connection to an ATE System

**Note:** *all control signals are optoisolated. You must power the optoisolation circuitry through the OPTO_5V and OPTO_GND lines of the "CONTROL" connector.*

*If your system doesn't require optoisolation, just connect the OPTO_5V and OPTO_GND lines of the "CONTROL" connector to the 5V and GND lines (respectively) of the "POWER" connector.*

# 4 Technical Specifications

## 4.1 Absolute Maximum Ratings

Table 4.1: Absolute Maximum Ratings

| Parameter | Value |
|---|---|
| **"CONTROL" Connector** | |
| Maximum input voltage on lines START, STOP, SEL[4..0], RX | -2V to +9V |
| Maximum input voltage on line RX_RS232 | -25V to +25V |
| Maximum input voltage on line OPTO_5V (reference OPTO_GND) | 6V |
| Maximum current on lines BUSY, PASS,FAIL,TX | -50mA to 1.5mA |
| Maximum current on line TX_RS232 | ±60mA |
| **"ISP" Connector** | |
| Maximum input voltage on lines DIO/AO[1..0], DIO[6..2], CLKOUT | -1V to +7V |
| Maximum input voltage on line AIN0 | -12V to +40V |
| Maximum current on lines DIO/AO[1..0], DIO[6..2], CLKOUT | ±50mA |
| Maximum current on line VPROG0 | 500mA |
| Maximum current on line VPROG1 | 250mA |
| **"POWER" Connector** | |
| Maximum supply voltage on line POWER (reference GND) | -20V to +30V |
| Maximum current on line 5V | 150mA |

# 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 lines START, STOP, SEL[4..0], RX | The driver must be able to provide at least 5mA | 0V | - | 2V |
| $V_{IH}$ (input high voltage) on lines START, STOP, SEL[4..0], RX | | 3V | - | 5V |
| $V_{IL}$ (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 lines BUSY, FAIL, PASS, TX | $I_{OL}$ = 4.5mA | - | - | 450mV |
| $V_{OH}$ (output high voltage) on lines BUSY, FAIL, PASS, TX | | 4.5V | - | 5V |
| $V_{OL}$ (output low voltage) on line TX_RS232 | $R_{LOAD}$ = 3KΩ | - | - | -5V |
| $V_{OH}$ (output high voltage) on line TX_RS232 | $R_{LOAD}$ = 3KΩ | +5V | - | - |
| OPTO_5V line power consumption | | - | 100mA | - |
| **"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 | - |
| $I_L$ (input leakage current) on line AIN0 | $V_{AIN0}$ = 25V | - | - | 4.3mA |
| AIN0 line input voltage | | 0V | - | 28.5V |
| VPROG0 line output voltage | | 1.6V | - | 5.5V |
| VPROG0 current (source) | | - | - | 500mA |
| VPROG1 line output voltage | | 3V | - | 14.5V |

| Parameter | Condition | Value | | |
|---|---|---|---|---|
| | | Min | Typ | Max |
| VPROG1 current (source) | | - | - | 250mA |
| **"POWER" Connector** | | | | |
| Supply voltage | | 9V | - | 24V |
| Power consumption | | - | - | 1.5A |
| 5V line output current | | - | - | 100mA |
| 5V line output voltage | | 4.75V | 5V | 5.25V |

# 4.3 AC Characteristics

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 | $V_{PROG1}$ = 3V | Load: 4.7KΩ//100pF (see figure 4.1a) | - | 7µs | - |
| | $V_{PROG1}$ = 12V | | - | 11µs | - |
| | $V_{PROG1}$ = 14.5V | | - | 12µs | - |
| $t_{FALL}$ on lines DIO/AO[1..0] configured as analog output | $V_{PROG1}$ = 3V | Load: 100pF (see figure 4.1b) | - | 8µs | - |
| | $V_{PROG1}$ = 12V | | - | 20µs | - |
| | $V_{PROG1}$ = 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 41b) | - | 300ms | - |
| | $V_{PROG0}$ = 3.3-0V | | - | 350ms | - |
| | $V_{PROG0}$ = 5.5-0V | | - | 350ms | - |
| $t_{RISE}$ on line VPROG1 | $V_{PROG1}$ = 0-3V | Load: 10Ω//1mF (see figure 4.1a) | - | 1.3ms | - |
| | $V_{PROG1}$ = 0-5V | Load: 47Ω//1mF (see figure 4.1a) | - | 1.8ms | - |
| | $V_{PROG1}$ = 0-14.5V | Load: 94Ω//1mF (see figure 4.1a) | - | 13ms | - |
| $t_{FALL}$ on line VPROG1 | $V_{PROG1}$ = 3-0V | Load: 1mF (see figure 4.1b) | - | 18ms | - |
| | $V_{PROG1}$ = 5-0V | | - | 30ms | - |
| | $V_{PROG1}$ = 14.5-0V | | - | 45ms | - |
| CLKOUT frequency | | | 0MHz | - | 50MHz |

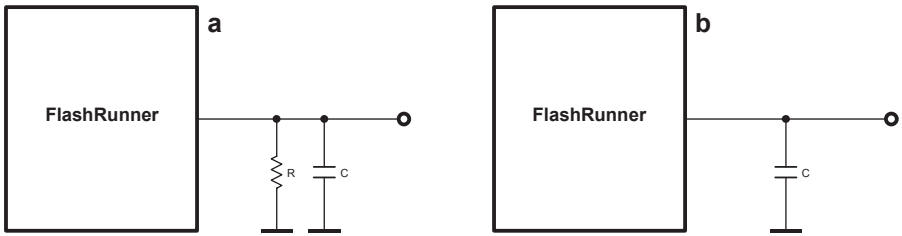



Figure 4.1: Load Conditions

## 4.4 Physical and Environmental Specifications

Table 4.4: Physical and Environmental Specifications

| Parameter | Value |
|---|---|
| Dimensions (with top panel) | 130 x 74 x 27 mm |
| Dimensions (without top panel) | 130 x 74 x 22 mm |
| “ISP” connector type | 36-pin, 2.54mm-pitch, dual-row header (male) |
| “CONTROL” connector type | 16-pin, 2.54mm-pitch, dual-row header (male) |
| “POWER” connector type | 20-pin, 2.54mm-pitch, dual-row header (male) |
| Operating temperature | 0-50°C |
| Operating humidity | 90% max (without condensation) |
| Storage temperature | 0-70°C |
| Storage humidity | 90% max (without condensation) |

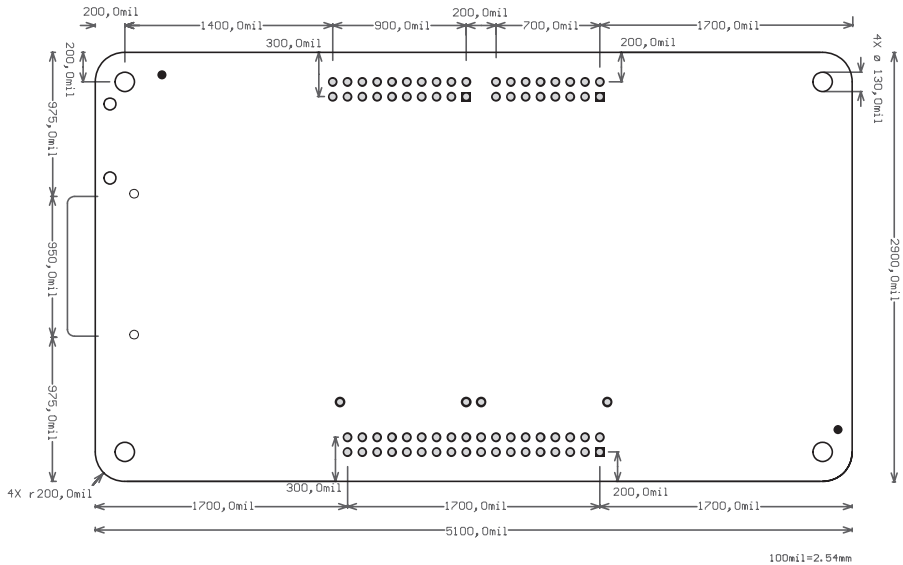



Figure 4.2: Bottom Layer (Connectors Side) Layout

SMH Technologies S.r.l.
Via Giovanni Agnelli, 1
33083 Villotta di Chions (PN) Italy

www.smh-tech.com
info@smh-tech.com