

R-Idge

6LoWPAN USB router

Firmware Installation

Manual

TABLE OF CONTENTS

I. INTRODUCTION	3
1.1 History and Credits	3
2. COMMAND LINE OPTIONS	4
2.1 Option Descriptions	4
2.2 Example Command Line Invocations	7
3. CONFIGURATION FILE	9
3.1 BLTOOL Defaults	9
3.2 Programmer Definitions	9
3.3 Part Definitions	10
3.4 Other Notes	10
A. PLATFORM DEPENDENT INFORMATION	11
A.1 Linux	11
A.1.1 Linux Installation	11
A.1.1.1 Installation from source files	11
A.1.1.2 RPM Package	11
A.1.1.3 DEB Package	12
A.1.2 Linux Configuration Files	12
A.1.3 Linux Port Names	12
A.1.4 Linux Documentation	12
B. TROUBLESHOOTING	13

I. INTRODUCTION

BLTOOL - Rosand Technologies Downloader Uploader - is a program for downloading and uploading code to ROSAND Technologies products.

BLTOOL can be used via the command line to read or write all chip memory types (flash, signature bytes). Using BLTOOL from the command line allows programming the entire memory of the chip from the contents of a file.

I.1 History and Credits

BLTOOL is derived from the program AVRDUDE written by Brian S. Dean. The texi formatting was modeled after that of the Simulavr documentation by Ted Roth.

2. COMMAND LINE OPTIONS

2.1 Option Descriptions

BLTOOL is a command line tool, used as follows:

```
bltool -p partno options ...
```

Command line options are used to control BLTOOL's behaviour. The following options are recognized:

-p *partno*

This is the only mandatory option and it tells BLTOOL what type of part (MCU) is used by the product. The *partno* parameter is the part's id listed in the configuration file. Specify -p ? to list all parts in the configuration file. If BLTOOL does not know about the part of the product that you have purchased, simply upgrade BLTOOL with a new version. Currently, the following MCU types are understood:

CC2531F128	MCU used by R-Idge
CC2531F256	MCU used by R-Mote Device

-c *programmer-id*

Specify the programmer to be used. The *programmer-id* parameter is the programmer's id listed in the configuration file. Specify -c ? to list all programmers in the configuration file. New programmer id will be added in the future to make BLTOOL work with different programmers. You can use the 'default_programmer' keyword in your `.bltoolrc` file to assign a default programmer to keep from having to specify this option on every invocation. Currently, the following programmer ids are understood and supported:

SWRA327	ROSAND Technologies Serial Programmer
---------	---------------------------------------

-C *config-file*

Use the specified config file for configuration data. This file contains all programmer and part definitions that BLTOOL knows about. If not specified, BLTOOL reads the configuration file from `/etc/bltool.conf` (Linux rpm and deb packages).

-D

Disable auto erase for flash. When the -U option with flash memory is specified, bltool will perform a chip erase before starting any of the programming operations, since it generally is a mistake to program the flash without performing an erase first. This option disables that.

-e

Causes a chip erase to be executed. This will reset the contents of the flash ROM to the value '0xff'. It is basically a prerequisite command before the flash ROM can be reprogrammed again. The only exception would be if the new contents would exclusively cause bits to be programmed from the value '1' to '0'.

-n

No-write - disables actually writing data to the MCU (useful for debugging BLTOOL).

-P *port*

Use *port* to identify the serial port to which the product is attached. By default the serial port `/dev/ttyACM0` will be used. If you need to use a different serial port, use this option to specify the alternate port name.

-q

Disable (or quell) output of the progress bar while reading or writing to the device. Specify it a second time for even quieter operation.

-v

Enable verbose output.

-U [memtype:op:]filename[:format]

Perform a memory operation. Multiple '-U' options can be specified in order to operate on multiple memories on the same command-line invocation. The *memtype* field specifies the memory type to operate on. Typically, a device's memory configuration at least contains the memory types *flash*. All memory types currently known are:

flash	The flash ROM of the device.
signature	The three device signature bytes (device ID + device info).

The *op* field specifies what operation to perform:

r	read the specified device memory and write to the specified file
w	read the specified file and write it to the specified device memory
v	read the specified device memory and the specified file and perform a verify operation

The *filename* field indicates the name of the file to read or write.

The *format* field is optional and contains the format of the file to read or write. Possible values are:

i	Intel Hex
s	Motorola S-record
r	raw binary; little-endian byte order, in the case of the flash ROM data
a	auto detect; valid for input only

The default is to use auto detection for input files, and raw binary format for output files.

Note that if *filename* contains a colon, the *format* field is no longer optional since the filename part following the colon would otherwise be misinterpreted as *format*.

As an abbreviation, the form **-U filename** is equivalent to specifying **-U flash:w:filename:a**. This will only work if *filename* does not have a colon in it.

-V

Disable automatic verify check when uploading data.

2.2 Example Command Line Invocations

Download the file `ridge-ver1_0.hex` to the R-Ildge USB Router connected to the default serial port:

```
% blttool -p CC2531F128 -c swra327 -e -U flash:w:ridge-ver1_0.hex

blttool: Device initialized and ready to accept instructions
Reading | ##### | 100% 0.00s

blttool: Device signature = 0xb54c07
blttool: NOTE: FLASH memory has been specified, an erase cycle will be
performed
        To disable this feature, specify the -D option.
blttool: erasing chip
blttool: reading input file "ridge-ver1_0.hex"
blttool: input file ridge-ver1_0.hex auto detected as Intel Hex
blttool: writing flash (96366 bytes):

Writing | ##### | 100% 7.50s

blttool: 96366 bytes of flash written
blttool: verifying flash memory against ridge-ver1_0.hex:
blttool: load data flash data from input file ridge-ver1_0.hex:
blttool: input file ridge-ver1_0.hex auto detected as Intel Hex
blttool: input file ridge-ver1_0.hex contains 96366 bytes
blttool: reading on-chip flash data:
blttool: input file ridge-ver1_0.hex auto detected as Intel Hex

Reading | ##### | 100% 1.84s

blttool: verifying ...
blttool: 96366 bytes of flash verified

blttool done. Thank you.
%
```

Upload the flash memory from the R-Mote device and save it in raw binary format in the file named `/tmp/flash.bin`:

```
% blttool -p CC2531F128 -c swra327 -U flash:r:"/tmp/flash.bin":r
blttool: Device initialized and ready to accept instructions
Reading | ##### | 100% 0.00s
blttool: Device signature = 0xb54c07
blttool: reading flash memory:
Reading | ##### | 100% 2.85s
blttool: writing output file "/tmp/flash.bin"
blttool done. Thank you.
%
```


3. CONFIGURATION FILE

BLTOOL reads a configuration file upon startup which describes all of the parts and programmers that it knows about.

BLTOOL first looks for a system wide configuration file in a platform dependent location. On Linux, this is usually `etc/bltool.conf`. The name of this file can be changed using the `'-C'` command line option.

After the system wide configuration file is parsed, BLTOOL looks for a per-user configuration file to augment or override the system wide defaults. On Linux, the per-user file is `.bltoolrc` within the user's home directory.

3.1 BLTOOL Defaults

```
default_serial = "default-serial-device";
```

Assign the default serial port device. Can be overridden using the `'-P'` option.

```
default_programmer = "default-programmer-id";
```

Assign the default programmer id. Can be overridden using the `'-c'` option.

3.2 Programmer Definitions

The format of the programmer definition is as follows:

```
programmer
  id      = <id1> [, <id2> [, <id3>] ...] ; # <idN> are quoted strings
  desc    = <description> ;                 # quoted string
  type    = SWRA327 ;                       # programmer type
;
```

3.3 Part Definitions

```
part
  id          = <id> ;           # quoted string
  desc        = <description> ;  # quoted string
  devicecode  = <num> ;          # numeric
  chip_erase_delay = <num> ;     # micro-seconds
  memory <memtype>
    paged     = <yes/no> ;       # yes / no
    size      = <num> ;          # bytes
    page_size = <num> ;          # bytes
    num_pages = <num> ;          # numeric
    block_size = <num> ;         # numeric
    min_write_delay = <num> ;   # micro-seconds
    max_write_delay = <num> ;   # micro-seconds
  ;
;
```

3.4 Other Notes

- Not all memory types will implement all instructions.
- Example memory types are: **flash**, **signature**.
- The memory type specified on the BLTOOL command line must match one of the memory types defined for the specified chip.

A. PLATFORM DEPENDENT INFORMATION

A.I Linux

A.I.1 Linux Installation

A.I.1.1 Installation from source files

You can download the source tarball from <http://www.rosand-tech.com/downloads>. To build and install from the source tarball on Unix like systems:

```
$ gunzip -c bltool-1.00.tar.gz | tar xf -
$ cd bltool-1.00
$ ./configure
$ make
$ su root -c 'make install'
```

The default location of the install is into `/usr/local` so you will need to be sure that `/usr/local/bin` is in your `PATH` environment variable.

If you do not have root access to your system, you can do the the following instead

```
$ gunzip -c bltool-1.00.tar.gz | tar xf -
$ cd bltool-1.00
$ ./configure --prefix=$HOME/local
$ make
$ make install
```

A.I.1.2 RPM Package

On rpm based Linux systems (such as RedHat, Fedora, SUSE, etc), you can download the rpm binaries from <http://www.rosand-tech.com/downloads>

```
$ su - root
# rpm bltool-1.00-001.i686.rpm
```

You can also build and install the rpm binaries directly from the tarball:

```
$ su - root
# rpmbuild -tb bltool-1.00.tar.gz
# rpm -Uvh /root/rpmbuild/RPMS/i686/bltool-1.00-1.i386.rpm
```

Note that the path to the resulting rpm package, differs from system to system. The above example is specific to Fedora.

A.1.1.3 DEB Package

On deb based Linux systems (such as Debian, Ubuntu, etc), you can download the deb package from <http://www.rosand-tech.com/downloads>.

```
$ su - root
# dpkg -i bltool_1.00-001_i386.deb
```

A.1.2 Linux Configuration Files

When BLTOOL is build from the source tarball using the default '--prefix' configure option, the default configuration file for a Linux system is located at `/usr/local/etc/bltool.conf`.

When CFGTOOL is installed from an rpm package, the system configuration file will always be `/etc/cfgtool.conf`.

When CFGTOOL is installed from a deb package, the system configuration file will always be `/etc/cfgtool.conf`.

The system default configuration file can be overridden by using the '-c' command line option. Additionally, the user's home directory is searched for a file named `.bltoolrc`, and if found, is used to augment the system default configuration file.

A.1.3 Linux Port Names

On Linux, the default serial port device is `/dev/ttyACM0`. BLTOOL uses the tty driver for serial port access.

A.1.4 Linux Documentation

BLTOOL installs a manual page as well as info, HTML and PDF documentation. The manual page is installed in `/usr/local/man/man1` area, while the HTML and PDF documentation is installed in `/usr/local/share/doc/bltool` directory. The info manual is installed in `/usr/local/info/bltool.info`.

Note that these locations can be altered by various configure options such as '--prefix'.

There are also rpm and deb documentation packages. They can be found at <http://www.rosand-tech.com/downloads>.

B. TROUBLESHOOTING

Please report any bugs encountered to <mailto:tech-support@rosand-tech.com>.