

SHDesigns Serial Downloader for
Z-World Rabbit Boards
Copyright (c) 2003-2006 SHDesigns

Date: Saturday, January 07, 2006

Current Version: 1.4

1.0 INTRODUCTION	1
2.0 HOW IT WORKS	1
3.0 IMPLEMENTING THE LIBRARY	2
3.1 Dynamic C Changes	2
3.2 User Program Changes	2
3.2.1 Define the port in your source file.	2
3.2.2 Dedicated direct connection	2
3.2.3 Shared serial connection	3
3.2.4 Modem serial loader	3
3.2.5 I/O Port Setup.	4
3.3 Including the RAM loader in FLASH	5
3.4 The Sample program	5
4.0 RAM DOWNLOAD PROGRAM	5
5.0 PC DOWNLOAD UTILITY	6
5.1 Installation	6
5.2 Requirements	7
5.3 Encrypted .bin files	7
5.4 Programming the Boards	7
5.4.1 Direct connection	7
5.4.2 Modem dial out	8
5.4.3 Modem dial in	8
5.5 Errors	8
6.0 ENCRYPTION	9

1.0 INTRODUCTION

The Serial Download Utility allows Rabbit-based boards to upgrade the FLASH code in the field without the serial programming cable. This functionality consists of 3 parts:

1. A small library included in a user program: SERDOWNL.LIB.
2. A small RAM-based program that downloads the new code and programs FLASH.
3. The PC Utility to find and program boards.

Unlike other solutions for serial downloading, this implementation requires little changes to a users code. There are no hardware changes or library changes.

There are three modes the software may be used:

1. Direct serial connection
2. Modem on Rabbit board that answers incoming call from Serial Download Program
3. Modem on Rabbit board that calls into the Serial Download Program

2.0 HOW IT WORKS

The Z-World solutions impose restrictions on the hardware and software. Their serial solutions require changes to the libraries and dividing FLASH in half.

The library uses a small RAM-based loader. This loader is small (24k.) The library routine has to allocate a buffer in xmem to store the program. After the code is downloaded and a 'RUN' command is received, it copies the program to root RAM, re-maps RAM to 0, and reboots to the new code.

The RAM loader may be downloaded from the PC or included in FLASH. Note for proper error recovery, the RAM loader should be included in FLASH.

The library also has additional functions. One is to preserve the serial parameters. Since the entire board is reset, the serial port and modem need to be reinitialized. These parameters are passed in RAM to the new program. There is also an option to pass Rabbit I/O port initialization to the RAM loader.

An function of the library is to respond to "Query" commands. The query command is a send by the PC utility to ask the board to identify itself. Each board that supports the download function will reply with a user-supplied string. This allows the PC utility to identify boards by name. The user can include the current version in the string to allow the PC user to identify boards that need upgrading.

The library uses no extra xmem memory until a request to download is received. It will then allocate enough xmem to store the RAM program. The user program is notified that a request has

been received and so it can free up xmem if needed. Only programs that use all of xmem would need to do this. If the RAM loader is included in FLASH, no extra xmem is needed.

The library and the download utility report their status to the PC program. They will report that they need the Ram loader, have it in FLASH, or are running it. Also, memory allocation and code blocks are acknowledged.

Note: Because the library runs a separate RAM program, the download process can not be run under the debugger. The Dynamic C IDE will reset the asnd halt board when the RAM loader is run.

3.0 IMPLEMENTING THE LIBRARY

3.1 Dynamic C Changes

Serdownl.lib needs to be added to the DC compiler LIB.DIR file for the compiler to locate the library. I.e. if the files are in “C:\program files\rabbit\mylibs” add the following lines to the LIB.DIR file in the main compiler directory:

```
C:\program files\rabbit\mylibs\serdownl.lib
```

For all the Rabbit-2k based boards there are no changes required for the Dynamic C libraries. This library has been compiled with version 7.25 of Dynamic C through 9.25. It should work with any 7.2x compiler or later.

As of 6-1-2003, separate code and data is supported and has been tested.

3.2 User Program Changes

In the user’s program the changes depend on how the serial loader is to be used. Each of the following sections describe the changes.

3.2.1 Define the port in your source file.

First define the serial functions to use:
This example uses serial port B

```
#define USE_SERIAL_B
```

if using I/O port D pins, add a define:
#define SERB_USEPORTD

In both defines above, replace ‘B’ with the port you wish to use.

3.2.2 Dedicated direct connection

In this case, the serial port is dedicated for serial download only. #define the USE_SERIAL_X to select the serial port to use.

Define DIRECT_SERIAL in your code before including the library:

```
#define DIRECT_SERIAL
```

Include serdownl.lib in your source:

```
#use serdownl.lib
```

In your main() function add the following:

A call to SerDL_Init() as follows:

```
SerDL_Init("string");
```

where "string" is the ID string to report to the PC. I.e. "ZYX Corp controller 1.0". This string will appear in the board list in the PC utility.

In your main loop of your program, call SerDL_Tick(). This function handles the download function.

3.2.3 Shared serial connection

In this case, the serial port is used for some other function until the user selects download. This requires that the loader be included in flash (see section 3.3.)

Include the library in your source:

```
#use serdownl.lib
```

When your user interface selects a download, call the following function:

```
SerLoad(0,baud,string,NULL,NULL);
```

Where the parameters are:

0 - first parameter of 0 specifies direct connection

baud - long integer specifying the baud rate.

string - An ascii string. This string will be reported to the PC utility.

3.2.4 Modem serial loader

The modem mode can either dial out to a PC running the download utility or answer a call from one. In this mode, the RAM loader must be included in FLASH (see section 3.3).

Include the library in your source:

```
#use serdownl.lib
```

When your user interface selects a download, call the following function:

```
SerLoad(int mode,long baud,char * string,char * modem_init,char * phone_number);
```

Where:

mode = 1 modem answer mode, 2= modem dial out

baud = baud rate

modem_init = string of commands to send to modem. Separate commands by \r
each command will be sent separately and wait for OK.

phone_number = string to dial, this is added to the ATDT command to the modem.
(not used for answer mode, use NULL);

3.2.5 I/O Port Setup.

Since the RAM loaders are precompiled, the I/O port setup could not be changed without compiling custom versions.. The RAM loaders and serdownl.lib dated 1/4/2006 or later implement a method to pass I/O port configuration. This allows devices to be powered up, modem status lines and driver enable pins to be initialized. The configuration is passed via the SERDL_IOINIT define:

```
#define SERDL_IOINIT {PDDR,0x1,PDDDR,0x1,PEDR,(char)~0x40,PEDDR,0x40,0}
```

If SERDL_IOINIT is defined, it is passed to the RAM loader. This is a list of Rabbit port and data to write. The above sets port D bit 0 as output high and port E bit 6 as an output low.

This should help with modems that need DTR/CTS etc set to work. It can also be used to power up serial drivers. Up to 16 pairs of ports/data can be defined. The following Rabbit internal I/O registers are supported:

```
GOCR, GCDR, MTCR, SPCR, PADR,  
PFDR, PFCR, PFFR, PFDCR, PFDDR,  
PBDR, PBDDR,  
PGDR, PGCR, PGFR, PGDCR, PGDDR,  
PCDR, PCFR, PDDR, PDCR, PDFR, PDDCR, PDDDR,  
PEDR, PECR, PEFR, PEDDR,  
PWL0R, PWM0R, PWL1R, PWM1R, PWL2R, PWM2R, PWL3R, PWM3R,  
TACSR, TAPR, TAT1R, TACR, TAT2R, TAT8R, TAT3R, TAT9R, TAT4R,  
TAT10R, TAT5R, TAT6R, TAT7R,  
SAAR, SACR, SAER, SECR, SEER, SBAR, SBCR, SBER,
```

SFCR, SFER, SCAR, SCCR, SCER, SDAR, SDCR, SDER,

3.3 Including the RAM loader in FLASH

The library can include the RAM loader in FLASH. This requires about 22k more of FLASH memory. To include the RAM loader add the following define before the “#use SERDOWNL.LIB line:

```
#define SERIAL_LOADER "x:\\path\\to\\file\\serdownl.bin"  
(see section 4.0 for the correct RAM loader to use.)
```

Note: The RAM loader implements recovery. This requires the RAM loader is included in flash for recovery to work.

The SERIAL_LOADER define is used in the library as follows:

```
#ximport SERIAL_LOADER _ser_dl_loader
```

The SERIAL_LOADER define is the file name to ximport. Dynamic C has problems with paths for ximport. Sometimes it will work with just the file name in quotes. But it often does not use the right directory. It is usually better to use the full path to the loader. Remember, backslashes need to be entered twice in ‘C’ as shown.

This file must not be encrypted.

3.4 The Sample program

In the Sample directory is a file called SerDLTest.c. This is basically a program that does nothing except allow a user program to be loaded. It will flash DS2 on the RCM22xx demo board.

Compile the program to FLASH and run it. Then run the PC utility. You should be able to download one of your regular .bin files to the board. This will then run your FLASH bin file. If your .bin file does not have the library functions included, you will not be able to download again.

Make the changes to your source as shown in the previous sections. Then once you program it to flash, any new version can be downloaded via the PC utility.

4.0 RAM DOWNLOAD PROGRAM

The RAM program is supplied pre-compiled. The pre-supplied loaders are compatible with most Rabbit boards. These are generic, the RAM loaders are not compiled for any specific board.

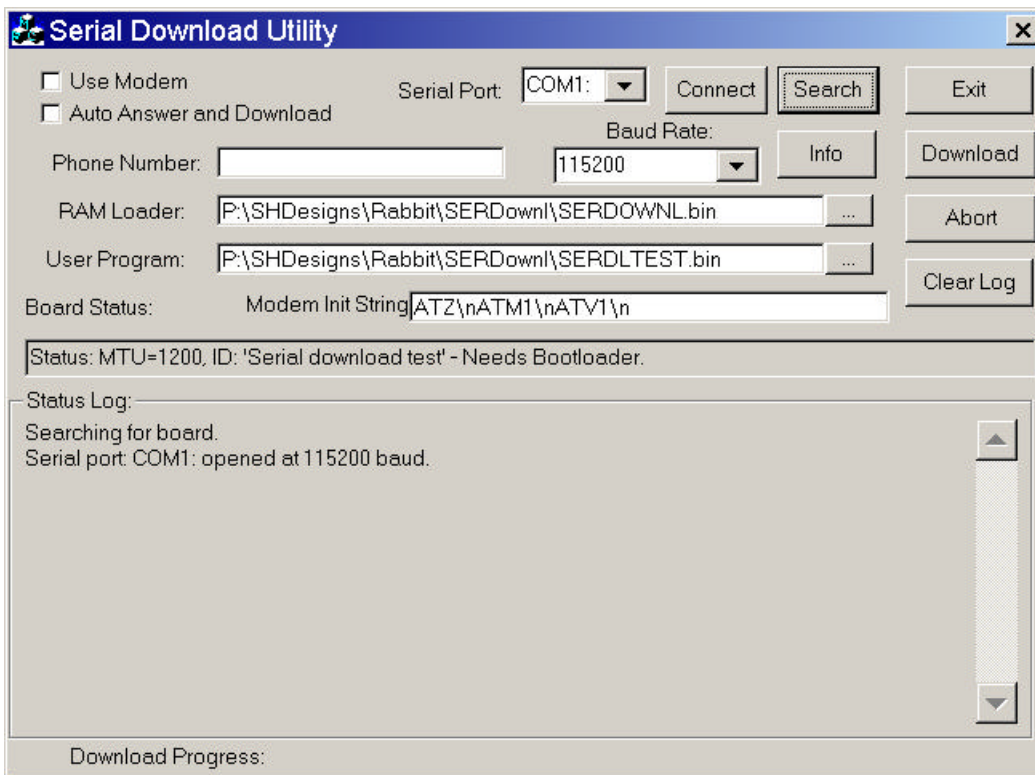
The file names are in the following format:
SDL-Generic-XY.bin

Where X= port A,B,C etc
Y=D if The alternate port D is to be used.

There are also versions with -Dbg in the name. These will output debug info on the programming port at 9600 baud. The "Diag" connector on the programming connector can be used with something like Hyperterm to capture debug output. If there are any other boards that you need, contact rabbit@shdesigns.org.

All the loaders are kept in the directory "Generic."

5.0 PC DOWNLOAD UTILITY



5.1 Installation

The PC utility is called SerDownload.exe. There is no installation needed. It can run from anywhere.

The only registry settings it uses is to store the last RAM loader file and user binary file locations and the port, modem settings. The next time the program is run, the file settings will default to the last used.

5.2 Requirements

Any PC can be used that runs windows. The program has also been run under Wine in Linux.

The PC Download utility requires either a properly-wired serial cable connected to the target or an internal or external modem.

5.3 Encrypted .bin files

The loader supports encrypted RAM and user program files. See the later section for details on how to encrypt the files. The encryption may include a password. When a password-protected file is downloaded, the user will be asked for the password. If the password matches, the download will take place. If an incorrect password is entered, the download is aborted.

An encrypted file may have no password. In this case, the user is not prompted for a password.

5.4 Programming the Boards

5.4.1 Direct connection

Select the serial port and baud rate. Then press on the “Connect” button. Make sure “Use modem” is not selected.

Click on search and the utility will poll the serial port for a board running the RAM loader or library. It should find the boards.

The board list should show the board found (there will only be one board in the list.) If none are found or the list needs to be updated, use the “Search” button.

At this point, the list should show the boards by name. They will have the “Needs Bootloader” status appended to the name. If the board includes the RAM loader in FLASH, the status will show "Has bootloader in XMEM FLASH."

Make sure to select the proper RAM loader (if needed by the board) and the user program bin file.

To download, press the download button.

The download will perform the following steps:

1. Send a request to the board for xmem to store the RAM loader.
2. Download the RAM loader (if needed.)
3. Send a “Run” command to start the RAM loader code.
4. Re-search for the board to find response from RAM loader.
5. Download the user code.

6. Send a “Reboot” command to run the user code.

Each step is acknowledged by the Rabbit board. A progress bar on the bottom of the window shows download completion.

Once the download is complete and the board has restarted, you should be able to perform a new search and see the board report the new status string (if new code has different ID string.)

5.4.2 Modem dial out

Enter the following settings:

1. Select “Use Modem”
2. Select the port to use
3. Set the Baud Rate
4. Enter the phone number to dial.
5. Enter the program .bin file
6. If needed change the modem init string.

Click on the “Connect” button. The program should dial out and connect to the Rabbit board. Once it is connected it will search for a response for the Ram loader.

Press “Download”. The board should be programmed.

5.4.3 Modem dial in

Enter the following settings:

1. Select “Use Modem”
2. Select the port to use
3. Set the Baud Rate
4. Select “Auto Answer and Download”
5. Enter the program .bin file.
6. If needed change the modem init string.

Click on the “Connect” button. The program will initialize the modem and wait for a connection from the rabbit board. When the Rabbit connects, it will automatically download the program.

5.5 Errors

The utility verifies each operation. It will try to recover from errors. The status log window will show these errors. The download can be retried if it fails.

Some notes:

1. Once the utility starts downloading the user FLASH code, the board is susceptible to a power loss or reset. If the board is reset in the middle of a download, it may not recover. In this

case the only way to fix the problem would be to use the Rabbit RFU utility and the serial debug cable.

2. If the user flash file is not found, the download will abort. At this point the RAM loader is running and the FLASH remains untouched. To recover, select the correct user program file and re-run the download. The utility will see that the board already has the RAM loader and skip right to the FLASH download.
3. If the Rabbit board does not answer or dial out a modem correctly, run the RAM loader directly in the DC IDE. Section 4.3 describes the details.

6.0 ENCRYPTION

The download utility version 1.1 and higher supports encrypted .bin files. This prevents users from using the bin files with any other downloader.

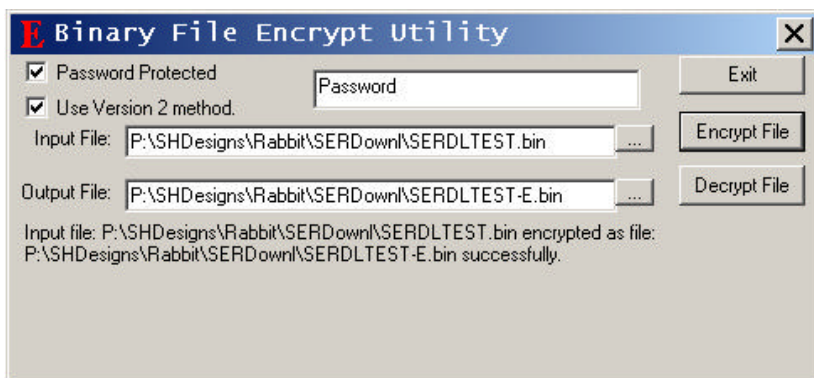
Encryption is done as follows:

1. A small header is added to the file
2. If a password is used, a encryption key is generated. If no password is used, a pre-defined key is used.
3. A second random key is generated.
4. This second key is encrypted with the password key.
5. The .bin file is appended and encrypted with the second key.

The password is not saved in the header. It is used to generate a unique key. There is no way to recover the password from the key. Thus, there is no information in the header on the size of the password. Unlike other encryption methods, the strength of the encryption does not depend on the length of the password.

The encryption keys are 96-bits long. The header starts with the string “Encrypted program file.” This identifies the file as encrypted. If a user types the file from a command prompt, they will see only this string.

A utility called EncryptBin.exe can be used to encrypt user files.



If the “Password Protected” check box, a password can be entered in the field to the right.

The input and output file fields can be entered or the “...” buttons on each can be used to browse for the files.

Pressing “Encrypt File” will encrypt the file. The status area in the bottom of the dialog will indicate a successful conversion.

Note: There is a difference between having no password and an empty password. If the “Password Protected” check box is not checked, the user will not be prompted for a password. If the box is checked and no password is entered, the user will still be prompted for a password and the loader will accept an empty password.

The “Decrypt file” button will decrypt a file, no password is needed.

Since this utility can decrypt a file with no password, it should not be distributed to end users.

For SSerDownload Versions 1.4 and later, the “Use Version 2 method” should be checked. This fixes a bug in earlier versions that encrypted files would accept multiple similar passwords.