Overview of the LTC2978 Configuration EEPROM (NVM)

Revision 1.0

Author: Mike Holloway, Rob Chiacchia, Linear Technology

Date: 11/11/2010

Revision History:

1.0	Added Arrow Contact Information
0.9	Added Instructions for ordering from Arrow
0.8	Added new diagrams on how to power the 2978/74
0.7	Update menu screenshots to agree with latest GUI
0.6	Updated BPM Micro section with new BP algorithm for serialization settings
0.5	Fixed Typos, Style. Added additional note boxes
0.4	Added additional references, note regarding terms and conditions
0.3	Referenced instructions for serialization in BPM Micro software.
0.2	Updated screenshots for BPM Micro programming option.
0.1	Initial Draft

Purpose

The LTC2978 provides a great deal of power and configurability for custom applications. The part additionally provides onboard non-volatile memory (NVM) or EEPROM to store and recall configuration parameters. This architecture allows the IC to power up and load the desired customer configuration **autonomously with no I^2C/firmware interaction required**.

A common question customers ask is, “Now that I have settled on a particular configuration, how do I program this configuration into the IC’s onboard non-volatile memory (NVM)”. A continuum of options is presented roughly in order of increasing complexity.

NOTE: Linear Technology strongly recommends **Option 1** below for pre-production prototyping, and **Option 2** or **Option 3 (or 3a)** for higher volume production. **Option 4** is a convenient method to program a small quantity of loose ICs prior to board assembly. **Option 5** requires a high level of software development expertise.

Options Involving the LTpowerPlay™ GUI

Most of the options presented here involve the use of Linear Technology's powerful LTpowerPlay™ GUI (available from http://ltpowerplay.com). In general, you will use the LTpowerPlay™ GUI and the DC1540A demonstration system to build a 'project' file for one or more ICs on your board. The demonstration system will allow you to verify that configuration parameters in the desired power supply management behavior. Once you have saved your configuration parameter into a project file (.proj file), you are ready to implement one of the following methods to transfer this configuration to the non-volatile memory (NVM) of the ICs on your board.

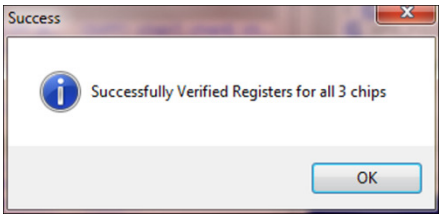
Option 1: Using LTpowerPlay™ to Program ICs on Customer Boards

The LTpowerPlay™ GUI is designed to communicate not only with LT-designed demo boards, but also with any number of LTC2978 ICs on a customer's board. Thus, you can use LTpowerPlay™ to program the parts on your board.

> **NOTE:** Linear Technology strongly recommends this option for pre-production prototyping, as it requires **no knowledge of I^2C** protocols and **no firmware development**. *This option also allows you to change the configuration on your board without de-soldering parts.*

The process is quite simple:

- Connect the LT USB-SMBus (DC1427 or DC1613) controller to your PC via USB, and to the target board (see datasheet, Figure 20. LTC Dongle Connections for Powering and Communicating with the LTC2978. This is shown below for convenience.)
- In the LTpowerPlay™ GUI:
    - Load your project (.proj) file using the "Open" button in the toolbar  .
    - Click the "Go Online" button in the toolbar  .
    - Click the "Write All" button in the toolbar  .
    - The GUI will program and VERIFY each operational register according to the project file. If this process is successful, you will see a message similar to this:



    - Click the "STORE_USER_ALL" button in the toolbar  .

All the LTC2978 ICs on your board now have the desired configuration stored into their NVM.

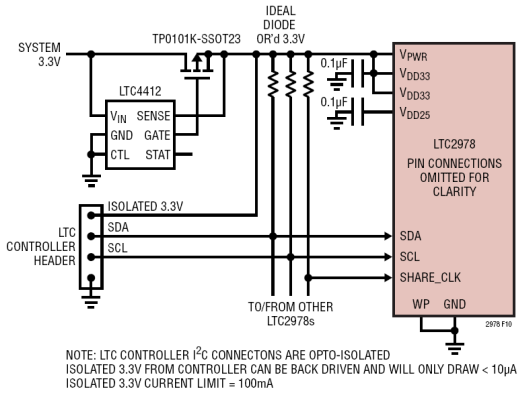
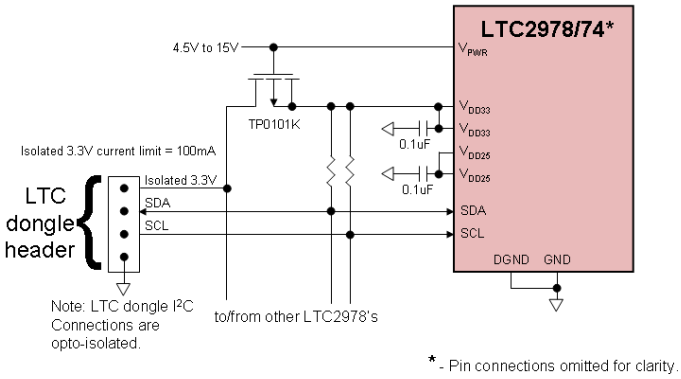


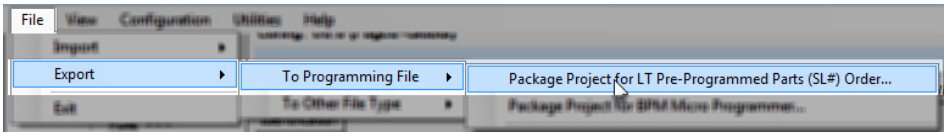
Figure 20. LTC Controller Connections for Powering and Communicating with the LTC2978



Option 2: Using LTpowerPlay™ to Order Pre-programmed Parts from Linear Technology

The second option, appropriate for high volume production, is to order pre-programmed parts from Linear Technology. Contact Linear Technology for minimum order quantities. Once you have used LTpowerPlay™ to save the project file (.proj) that you desire, follow these simple steps:

- In the LTpowerPlay™ GUI:
  - Load your project (.proj) file using the "Open" button in the toolbar  .
  - Under the menu, select "File, Export, To Programming File, Package Project for Pre-Programmed Parts (SL#) Order…"



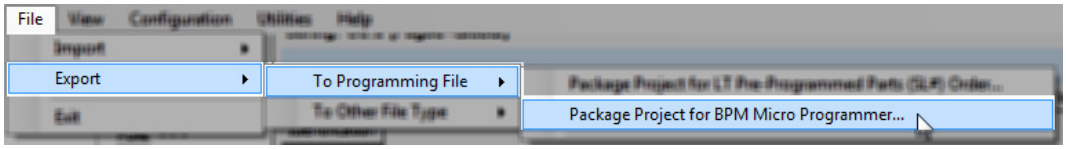
  - Browse for the name of the .zip file to write.
- The GUI will save a .zip file in the location you specified above.
- Send the .zip file to your Linear Technology Field Representative and you are ready to order pre-programmed parts.
- NOTE: If you are ordering more than one unique configuration, you need to clearly specify to Linear Technology the quantity required for each unique EEPROM CRC ID. See the instructions file included in the .ZIP file ('OrderingInstructions.txt') which lists each I^2C address (or U number) in your GUI project file to with its unique EEPROM CRC ID.

NOTE: If you choose to not purchase pre-programmed devices from LTC, terms and conditions will apply. Please check with your LTC Field Rep regarding device warranty if a Contract Manufacturer or a Third Party Programming House configures the LTC2978.

Option 3: Using LTpowerPlay™ and a BPM Micro Programmer to Create Custom Programmed Parts

Once you have the project file (.proj) that you desire, follow these simple steps to order pre-programmed parts from a contract manufacturer or a third party programming house that uses BPM Micro programmers:

- In the LTpowerPlay™ GUI:
    - Load your project (.proj) file using the "Open" button in the toolbar  .
    - Under the menu, select "File, Export, To Programming File, Package Project for BPM Micro Programmer…"



    - Browse for the name of the .zip file to store.
- The GUI will save a .zip file in the location you specified above. The .zip file contains a sub-folder for each IC in your project file.

    The .zip file is created for your convenience to contain the individual .oem hex files required for each IC in one file.
- Inside each sub-folder of the .zip file is a separate OEM Programming Hex (.oem) file for the corresponding IC on your board.
    - Send this .oem hex file to your contract manufacturer.
    - Your contract manufacturer will use this .oem file to program a number of ICs with the desired configuration.
    - Repeat this step for the individual .oem file included in the .zip file for each of the remaining ICs in your project.

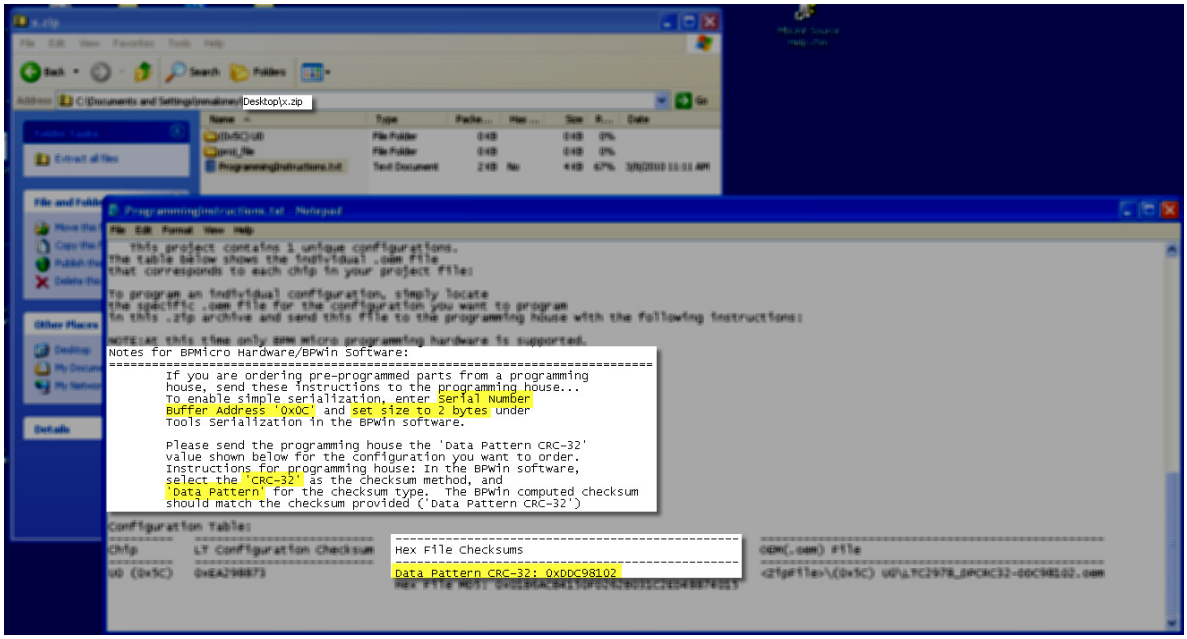
The contract manufacturer requires two pieces of hardware to program the LTC2978:

- A BPM Micro programmer (BPM Micro supports their full line of programmers for the LTC2978)
- A socket module for the LTC2978

Consult http://www.bpmicro.com for further details

NOTE: BPM Micro programmers use the BPWin software to load data patterns and program devices. Customers and contract manufacturers commonly use a computed 'checksum' to uniquely identify a configuration. The LTpowerPlay pre-computes a checksum for you that BPWin can also to use to identify a configuration. More detail is included in the .zip file under the 'ProgrammingInstructions.txt' file.

Open the .zip file generated above, and look at the text file 'ProgrammingInstructions.txt' in the main folder. This text file includes detailed ordering instructions for each of the ICs in your project file. The checksum that can be used to communicate with your contract manufacturer is called the 'Data Pattern CRC-32', and this checksum is computed and listed individually for each IC in your project file in the 'ProgrammingInstructions.txt' file as shown below.



The highlighted instructions above excerpted from 'ProgrammingInstructions.txt' explain how to:

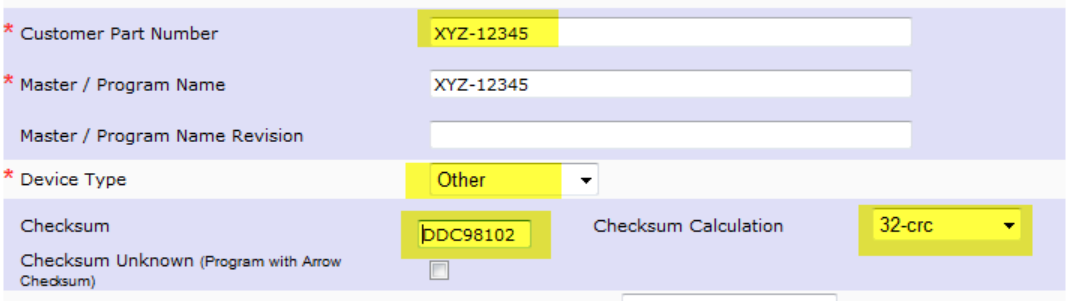
- Configure the BPWin software to compute the '**Data Pattern CRC-32'** checksum
- Configure the BPWin software for simple serialization of devices (serial number programming)

> **NOTE:** On 6/21/2010 BPM Micro updated their BETA programming algorithms to correct an issue with simple serialization. You need to update to the latest BPWin software version if you are using the serialization feature.
>
> As before, the serial number is always stored as little endian in the data pattern (address 0xC holds the LSB and address 0x0D holds the MSB of the 16-bit serial number). However, if you are using simple serialization, it is now necessary to additionally check the box titled '**reverse byte order'** as the BPWin software now instructs you when you select the LTC2978 device.
>
> The ProgrammingInstructions.txt file generated by the LTpowerPlay GUI does not yet include this additional instruction. In the future, the GUI will be updated to include this information in ProgrammingInstructions.txt.

Option 3a: Using Arrow Programming Services

Arrow utilizes the BPM programming hardware/software. Linear Technology has qualified Arrow as a provider of pre-programmed LTC2978 parts. If you choose Arrow, please follow these steps to order your pre-programmed parts"

- First, follow the instructions in Section 3 to generate individual an .oem hex file for each unique configuration you are going to order from Arrow. As mentioned in Section 3, this process will generate a .zip file with each individual .oem hex file and a ProgrammingInstructions.txt file you will need to complete your orders. Use the files within the .zip archive as indicated below.
- For each unique configuration:
    - Fill out Arrow's pre-programming worksheet for your First Article order (http://app.arrownac.com/ppw)



    - On this web form, be sure to fill in the following:
        - Enter a unique **customer part number** that corresponds to this configuration.
        - Select Device Type: **Other**
        - Enter the unique 'Data Pattern CRC-32' value corresponding to the .oem hex file for this configuration (consult ProgrammingInstructions.txt in the .zip archive which lists the Data Pattern CRC for each .oem hex file)
        - Select Checksum Calculation method: **32-crc.**
        - Fill in the remaining information on the form (contact Arrow Global Programming Services at (775) 334-1000 or email: p4FirstArt@arrow.com if you have questions)
    - Click Submit.
- You will then be emailed a confirmation email for this order. Use the link provided in the email to attach/upload the corresponding .oem hex file that corresponds to this configuration.
- Contact Arrow Global Programming Services at (775) 334-1000 or email: p4FirstArt@arrow.com for further questions on their pre-programmed parts service.

Option 4: Use LTpowerPlay™ to Program Loose ICs with the DC1508

The LTpowerPlay™ GUI can also program individual "loose ICs" using the DC1508 socketed programming board. This may be an appropriate option if you are unable to physically connect the LTpowerPlay™ GUI to your target board (ala Option 1). If this is the case, you can program individual "loose ICs" before soldering them down on your board.

> **NOTE: Option 1** is strongly preferred over this option as it allows you to change your configuration at will without de-soldering parts.

The process for programming individual loose ICs with the DC1508 board is as follows:

- Connect the DC1613 (LT USB-SMBus controller) to your PC, via USB, and to the DC1508 board.
- In the LTpowerPlay™ GUI:
    - Load your original desired project (.proj) file using the "Open" button in the toolbar  .
    - Remove all ICs except the first IC in the project
        - Select the IC to delete and press the "Delete Chip" button in the toolbar  .
    - Save this modified single-IC project to a new project file (i.e. IC1.proj) using the save button in the toolbar  .
    - Click the "Go Online" button in the toolbar  .
    - Click the "Write All' button in the toolbar  .
    - Click the "STORE_USER_ALL" button in the toolbar  .
    - Repeat the above steps for the remaining ICs in the original project file.

Option 5: Author Custom Firmware

NOTE: Linear Technology strongly recommends the use of one of the above options over authoring your own custom firmware. In general, it is not cost effective to develop custom firmware to load and store all the configuration registers.

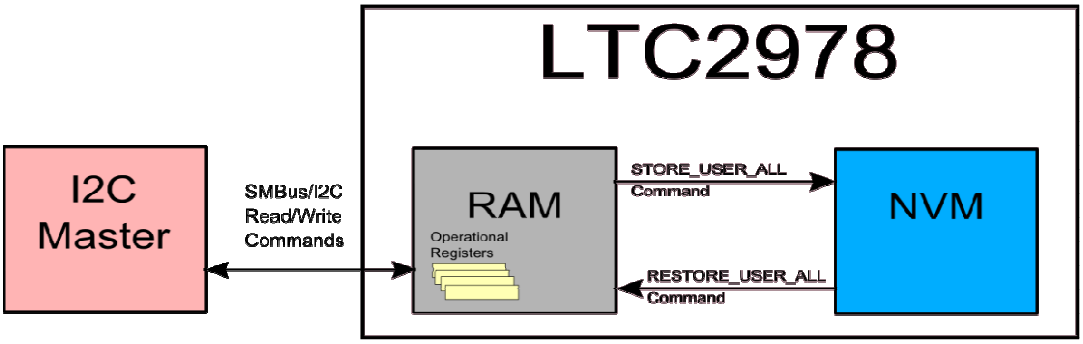
For high volume applications, email appssupport@ltpowerplay.com to request further information on in-system programming.

As shown in 'Appendix A: Memory Architecture', storing a configuration to non volatile memory (NVM) involves writing individual registers via the appropriate I^2C write commands and finally issuing a STORE_USER_ALL command to store this configuration into NVM.

Customers are free to implement their own firmware to communicate with the LTC2978 over the I^2C bus and do just that. However, this is the most difficult option because it involves detailed knowledge of the IC's register set, interaction of the various commands (i.e. WRITE_PROTECT), I^2C protocols, and an investment in authoring firmware.

Appendix A: Memory Architecture

The LTC2978 follows the standard PMBus model for configuration memory, implementing both operational registers (RAM) and non volatile memory (NVM). The following diagram illustrates the configuration memory architecture:



The operational RAM contains a number of 'registers' that control the operation of the IC. These operational registers can be read or modified using simple PMBus/SMBus/I^2C read/write commands. Consult the 'Register Command Set' table in the datasheet for a comprehensive list of these commands.

The IC also contains on-board non volatile memory (NVM) that is used to store the desired value of these operational registers across power cycles. Once the desired configuration has been programmed into RAM, storing the configuration to NVM is achieved with a single command called, 'STORE_USER_ALL' (command code 0x15). When the IC receives the STORE_USER_ALL command, it copies all the registers that are persisted to NVM from the operational RAM to the IC's onboard NVM for persistent storage. Consult the 'Register Command Set' table in the datasheet for a full list of these registers.

Each time the device powers up and also when a RESTORE_USER_ALL command is received, the IC transfers the configuration values stored in its onboard NVM into its operational registers.

This architecture allows the IC to power up and load the desired customer configuration autonomously with no I^2C/firmware interaction required.