Writing Configuration Files for Intersil Digital Power

Introduction

Intersil Digital-DC™ devices must be configured through pin-strap settings or by using PMBus™ commands. A configuration file is a human-readable text file that contains a sequence of PMBus commands to be written to a device. Configuration files also aid in sharing device settings to others for additional development, troubleshooting, or manufacturing.

This document covers how to write configuration files in the correct format, provides guidelines on structuring configuration files for the purpose of saving items into non-volatile storage, and protecting parameters via password protection. The process for creating a configuration file is quite simple and is covered in the first few pages of this document. The ability to add password protection is discussed beginning in Appendix A.

This application note should be used in conjunction with other Intersil application notes as listed in the Reference section, as well as the PMBus Power System Management Protocol Specification (referred to as PMBus Specification), to serve as a reference on command names and formats.

Configuration File Format

Configuration files are text files that can easily be edited using a text editor such as Microsoft Notepad. A command written in a configuration file consists of a PMBus command, followed by whitespace in the form of <tab> or <space> characters. Following the white space is the data for that command. The separation of commands is dictated by a carriage return character; therefore every command must begin at the beginning of a new line.

One feature of the configuration file format is that the data written can be written in different formats depending on the context of the command. As a default, any command's data can be written in hexadecimal format, as long as the hex data is preceded by a "0x", as shown in Figure 1. Note that when writing the data in hex, the values are in the more natural form of MSB to LSB, even though the

command data is physically sent in the reverse order as described in the PMBus specification, Part I.

The first data feature is for commands using the linear format or linear mode floating point format. These can be written using a floating point number. The units for these commands are the same as described in the PMBus specification for that particular command. An example of this is in Figure 2.

Second, for commands that are typically used for ASCII data, the data can be entered in as an ASCII string instead of in hex. This feature applies to the manufacturer commands (e.g. MFR_ID) in the PMBus spec, as well as the password commands used on Intersil devices. An example of using this is shown in Figure 3. Please note that once ASCII data is typed on a line, all characters typed on that line, even whitespace, will be interpreted as ASCII data to be written until a new line is introduced.

Lastly, for Intersil devices, the PID_TAPS command data may be configured using floating point values as long as they follow the format demonstrated in Figure 4.

```
VOUT_COMMAND . . . . . . . . . . . . . . . . . . . . . . . . . . . 0x699A
```

FIGURE 1. WRITING COMMAND DATA IN HEX

```
VOUT_COMMAND . . . . . . . . . . . . . . . . . . . . . . . . . . . 3.3 #Volts
```

FIGURE 2. WRITING COMMAND DATA IN FLOATING POINT

```
MFR_ID. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .Example OEM
```

FIGURE 3. WRITING COMMAND DATA IN ASCII

```
PID_TAPS . . . . . . . . . . . . . .  A = 1634, B = -2799, C = 1227
```

FIGURE 4. WRITING PID TAPS AS FLOATING POINT VALUES

Comments can also be added to a configuration file by preceding the comments with a pound ("#") character. Comments can be placed between command lines, and after command data, except for data written as ASCII characters.

Configuration File Structure

This section explains how to structure configuration files in order to store settings into the non-volatile memory.

Structuring for Command Storage

Aside from following the formatting guidelines, it is important to write a configuration file in an order that performs storage and command protection operations correctly. First, decide where to store the device settings. Intersil devices offer storage of commands in the Default Store, and many of the devices also have a User Store. The Default Store is typically used for keeping commands an OEM or Module Maker wants to keep as "default" settings that a user can always revert to by performing a RESTORE_DEFAULT_ALL. The User Store is typically used to let a user store additional settings outside of the default settings, and/or make changes that overlap the default settings to better suit their needs. More information about the Default and User stores can be found in Section 6 of the PMBus Specification, Part II.

When writing a configuration file with the intent of storing settings into a store, the following procedure must be performed in order to store the commands successfully. Note that the "XXXX" used below pertains to either "USER" or "DEFAULT", depending on which store is used.

1. Clear the store of any of its previous settings by writing a RESTORE_FACTORY and then a STORE_XXXX_ALL.
2. Perform a RESTORE_XXXX_ALL to prepare the device for adding commands.
3. Write the desired settings into the device that are to be stored in the XXXX store.
4. Write a STORE_XXXX_ALL to store the settings.

This procedure is demonstrated in Figure 5, along with a complete example of storing data into User & Default Stores in Figure 6. To add password protection to a configuration, see the password protection guidelines in Appendix A.

```
RESTORE_FACTORY  # Clear
STORE_XXXX_ALL   # XXXX Store

RESTORE_XXXX_ALL # Prepare XXXX
                 # for adding cmds

# Insert configuration data
# you want in XXXX Store

VOUT_COMMAND         3.3 #Volts
PID_TAPS     A=1634, B=-2799, C=1227

STORE_XXXX_ALL   # Store Settings
```

FIGURE 5. GENERAL STRUCTURE FOR STORING PARAMETERS INTO NON-VOLATILE MEMORY

```
# Perform actions for Default Store

RESTORE_FACTORY     # Clear
STORE_DEFAULT_ALL   # Default Store
STORE_USER_ALL   # User Store

RESTORE_DEFAULT_ALL # Prepare Default
                    # for adding cmds

 # Insert configuration data
 # you want in Default Store

MFR_ID              Example OEM
VOUT_MAX            5.0 #Volts
VOUT_COMMAND        3.3 #Volts

STORE_DEFAULT_ALL   # Store Settings

# Perform actions for User Store


RESTORE_USER_ALL # Prepare User
                 # for adding cmds

 # Insert configuration data
 # you want in User Store

VOUT_COMMAND        2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

STORE_USER_ALL      # Store Settings
```

FIGURE 6. STORING VALUES IN THE DEFAULT AND USER STORES

Standard Format for Configuration File

This section describes the recommended format for configuration text files, including command grouping, change logs, and related text comments. This format enables the creation of a consistent, traceable directory of project files which can simplify quality control from development through production. The command sections listed are relationally grouped to simplify development as well as for readability. This example text config file is shown in two column format for use in this document; normally the file is a single column of text lines. Some commands may not be required in all applications. Many device commands that could be listed in this document were omitted for brevity. Always consult the appropriate device data sheets and application notes during configuration file development.

File name of config file

<Project/BoardName>_<DeviceAddr>_<RailName/No.>_<DeviceNo.>_<FileRev>.txt

File format

#————————————————————

#This configuration file is intended for the device #described

in the filename of this file and the ASCII #MFR_xxxx commands in this file

#All PASSWORD protections must be cleared on the #device before loading this file

#

#Device ID: <DEVICE_ID>

#Schematic revision: <schematic revision level>

#BOM revision: <BOM revision level >

#PowerNavigator Revision: <GUI revision>

#Revision Log:

#Rev. x.x <date>, <author>

a) <log>

b) <log>

#Rev. x.x-1 <date>, <author>

a) <log>

b) <log>

#————————————————————

#Configuration File Line Syntax:

#PMBus Command <tab> Hex Value

#Erase user store & default store

RESTORE_FACTORY

STORE_USER_ALL

STORE_DEFAULT_ALL

#Prepare device for all commands to be added to the DEFAULT store

RESTORE_DEFAULT_ALL

#Manufacturer information fields in ASCII

#The sum total of ASCII characters for all #MFR_xxxx commands must be less than 128 char's

#MFR_SERIAL . . . . . reserved for time of manufacturing

#MFR_DATE . . . . . reserved for time of manufacturing

MFR_ID . . . . . <Company Name or Project Name>

MFR_MODEL . . . . . <Rail Name or Board in Project>

MFR_LOCATION . . . . . <Location or Ref Designator>

MFR_REVISION . . . . . <Revision of this Config File>

#Output Voltage commands

VOUT_COMMAND . . . . . <nn>

Vout Margin values (if different than factory %)

Vout Fault Thresholds (if different than factory %)

POWER_GOOD_ON

PG_DELAY

Vout Fault Responses (OV/UV)

OVUV_CONFIG

Output current

IOUT_CAL_GAIN . . . . . <nn>

IOUT_CAL_OFFSET . . . . . <nn>

Over current fault thresholds (peak and average)

Under current fault thresholds (peak and average)

Over current fault response

Under current fault response

#Input Voltage

Vin fault thresholds

Vin fault responses

#Other Faults

Temperature fault threshold

Temperature fault response

VMON fault threshold (in applicable devices)

VMON fault response (in applicable devices)

#General converter commands

TON_DELAY . . . . . <nn>

TON_RISE . . . . . <nn>

TOFF_DELAY . . . . . <nn>

TOFF_FALL. . . . . <nn>

FREQUENCY_SWITCH. . . . . <nn>

PID_TAPS. . . . . <nn nn nn>

MAX_DUTY . . . . . <nn>

DEADTIME . . . . . <nn>

DEADTIME_CONFIG . . . . . <nn>

INDUCTOR . . . . . <nn>

ON_OFF_CONFIG . . . . . <nn>

#Advanced commands

USER_CONFIG. . . . . <nn>

MFR_CONFIG . . . . . <nn>

```
NLR_CONFIG ..........................................<nn>
TRACK_CONFIG ........................................<nn>
MISC_CONFIG .........................................<nn>
INTERLEAVE ..........................................<nn>
DDC_CONFIG ..........................<nn>  #Set rail no.
SEQUENCE ............................................<nn>
TEMPCO_CONFIG .......................................<nn>
XTEMP_SCALE .........................................<nn>
XTEMP_OFFSET ........................................<nn>

#Security Settings
PUBLIC_PASSWORD ..............................  <xxxx>
PRIVATE_PASSWORD ..........................<yyyyyyyyy>
UNPROTECT ............................<nnnnnnnnnnnn>

# comment well !

STORE_DEFAULT_ALL
RESTORE_DEFAULT_ALL#comment out if USER stores follow this
command

# - end of file -
```

Modifying a Configuration File Saved from PowerNavigator

Instead of writing a new configuration file for each rail or application, many users will find it easier to modify a configuration file saved from the PowerNavigator software. Configuration files generated from PowerNavigator don't require users to look through command documentation in order to achieve the right settings, especially for bit-field commands when settings are only set via hex values. However, in order to add elements such as password protection to a config file saved in PowerNavigator, the file will need to be modified. This section goes through the process of saving and modifying a configuration file, and explains the structure of files saved in PowerNavigator.

Step 1: Use the PowerNavigator software to set the various settings as desired. In this example, a ZL2005 has the following commands set via the following steps:

1. Open the PowerNavigator software, and switch to "Device Config" mode.
2. Clear device by going to the PMBus Commands -> PMBus:Store tab, then clicking in order RESTORE_FACTORY, STORE_DEFAULT_ALL, STORE_USER_ALL. Note that no password protection can be present in order for this step to work.
3. Click the RESTORE_DEFAULT_ALL button to bring Default Store settings to active memory. Now select the desired settings you want in the Default Store (except for passwords). In this example, the commands set are VOUT_COMMAND & VOUT_MAX (found on the PMBus: Basic tab), as well as MFR_ID & MFR_SERIAL (found on the PMBus: Store tab).
4. After setting the desired settings for the Default Store, go back to the PMBus: Store tab, and click the STORE_DEFAULT_ALL button to store the command values.
5. Click the RESTORE_USER_ALL button to bring User Store settings to active memory. Now select the desired settings you want in the User Store (except for passwords). In this example, the commands set are VOUT_COMMAND & PID_TAPS (found on the PMBUS:Basic tab).
6. After setting the desired settings for the User Store, go back to the PMBus: Store tab, and click the STORE_USER_ALL button to store the command values.
7. Go to the "File I/O" tab, and proceed to create a configuration file, as shown in Figure 7. The process of saving a configuration file involves saving any data left unsaved into the User Store (or Default Store if User Store does not exist). It will also temporarily restore Default and Factory settings to active memory. Because of this, it is advised that the part be disconnected from any load to ensure that no damage occurs during the save process.

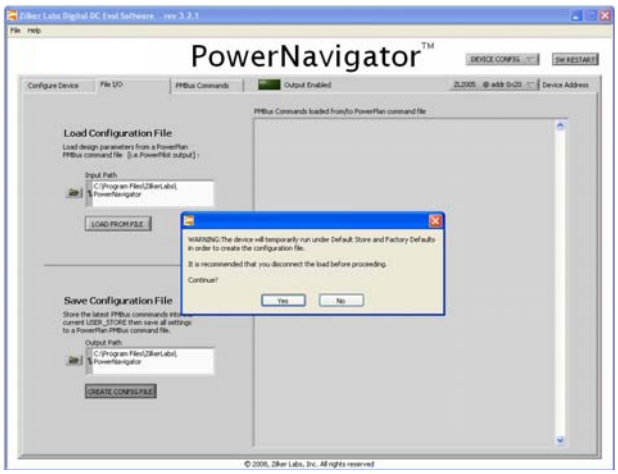


FIGURE 7. A SCREENSHOT OF SAVING A CONFIGURATION FILE IN POWERNAVIGATOR

Step 2: After saving the configuration file, open it in a text editor (e.g. Microsoft Notepad). Upon opening, the configuration file should look similar to Figure 8. As seen in this figure, there is a long list of Factory Default settings, all of which are commented out. This section is for informational use only, but should be checked to see that the pin-strap / hardcoded values in the device are acceptable, and will still be acceptable when the device is used in a production circuit (which might have different pin-strap settings).

```
# Configuration file for ZL2005-002-DC21  at Device Address 0x20
# Created on: Tue Aug 19 17:05:57 2008


#-------------------------------------------------------
# Clear Memory
# WARNING: Make sure no Password Protection
#                    is set before loading this
#                    config file.
#-------------------------------------------------------
RESTORE_FACTORY
STORE_DEFAULT_ALL
STORE_USER_ALL

#-------------------------------------------------------
# Factory Settings (for informational use only)
#-------------------------------------------------------

#Commands determined by pins V0:V1
#VOUT_COMMAND                    5.000122
#VOUT_COMMAND                     0xA001

#VOUT_MAX                        5.500000
#VOUT_MAX                         0xB000

#VOUT_MARGIN_HIGH                5.250000
#VOUT_MARGIN_HIGH                 0xA800

#VOUT_MARGIN_LOW                 4.750122
#VOUT_MARGIN_LOW                  0x9801

#VOUT_OV_FAULT_LIMIT             5.750244
#VOUT_OV_FAULT_LIMIT              0xB802

#VOUT_UV_FAULT_LIMIT             4.249878
#VOUT_UV_FAULT_LIMIT              0x87FF

#POWER_GOOD_ON                   4.500122
#POWER_GOOD_ON                    0x9001

#Commands determined by the UVLO pin
#VIN_UV_FAULT_LIMIT              4.500000
#VIN_UV_FAULT_LIMIT               0xCA40

#VIN_UV_WARN_LIMIT               4.640625
#VIN_UV_WARN_LIMIT                0xCA52

#Commands determined by pins ILIM0:ILIM1
#IOUT_OC_FAULT_LIMIT             30.000000
#IOUT_OC_FAULT_LIMIT              0xDBC0

#IOUT_UC_FAULT_LIMIT             -30.000000
#IOUT_UC_FAULT_LIMIT              0xDC40

#MFR_CONFIG                       0xAA01

#IOUT_AVG_OC_FAULT_LIMIT         30.000000
#IOUT_AVG_OC_FAULT_LIMIT          0xDBC0

#IOUT_AVG_UC_FAULT_LIMIT         -30.000000

                                  (continued on next page)
```

```
                          (continued from previous page)

#Commands determined by pins SA0:SA1
#INTERLEAVE                       0x0100

#-------------------------------------------------------
# Default Store Data
#-------------------------------------------------------

# The next line is required to insert Default Store  parameters
RESTORE_DEFAULT_ALL
#VOUT_COMMAND                    3.300049
VOUT_COMMAND                     0x699A

#VOUT_MAX                        5.000000
VOUT_MAX                         0xA000

MFR_ID                           Example OEM

MFR_SERIAL                       SSSNNN


STORE_DEFAULT_ALL  #uncomment to store above settings

#-------------------------------------------------------
# Current Settings/User Store Data
#-------------------------------------------------------

# The next line is required to insert User Store  parameters
RESTORE_USER_ALL
#VOUT_COMMAND                    2.500000
VOUT_COMMAND                     0x5000

#PID_TAPS                        A=1634.000000, B=-2799.000000, C=1227.000000
PID_TAPS                         A=0x7BCC40, B=0xFCAEF0, C=0x7B9960


STORE_USER_ALL  #uncomment to store above settings


#-------------------------------------------------------
# Soft Reset of Device
#-------------------------------------------------------
RESTORE_FACTORY
RESTORE_DEFAULT_ALL
RESTORE_USER_ALL
```

FIGURE 8. A CONFIGURATION FILE GENERATED BY POWERNAVIGATOR

Note: Some commands from the Factory Defaults section have been removed for printing purposes.

Step 3: After determining that the factory values are acceptable, you may want to remove them from the configuration file. Doing so will lead to a configuration file appearing as in Figure 8.

```
# Configuration file for ZL2005-002-DC21  at Device Address 0x20 MODIFIED

#-------------------------------------------------------
# Clear Memory
# WARNING: Make sure no Password Protection
#                    is set before loading this
#                    config file.
#-------------------------------------------------------
RESTORE_FACTORY
STORE_DEFAULT_ALL
STORE_USER_ALL

#-------------------------------------------------------
# Default Store Data
#-------------------------------------------------------
# The next line is required to insert Default Store  parameters
RESTORE_DEFAULT_ALL
#VOUT_COMMAND                    3.300049
VOUT_COMMAND                     0x699A

#VOUT_MAX                        5.000000
VOUT_MAX                         0xA000

MFR_ID                           Example OEM

MFR_SERIAL                       SSSNNN

STORE_DEFAULT_ALL  #uncomment to store above settings

#-------------------------------------------------------
# Current Settings/User Store Data
#-------------------------------------------------------
# The next line is required to insert User Store  parameters
RESTORE_USER_ALL
#VOUT_COMMAND                    2.500000
VOUT_COMMAND                     0x5000

#PID_TAPS                        A=1634.000000, B=-2799.000000, C=1227.000000
PID_TAPS                         A=0x7BCC40, B=0xFCAEF0, C=0x7B9960

STORE_USER_ALL  #uncomment to store above settings

#-------------------------------------------------------
# Soft Reset of Device
#-------------------------------------------------------
RESTORE_FACTORY
RESTORE_DEFAULT_ALL
RESTORE_USER_ALL
```

FIGURE 9. A CONFIGURATION FILE GENERATED BY POWERNAVIGATOR, WITH FACTORY SETTING INFORMATION REMOVED

Appendix A: Adding Password Protection Guidelines

Aside from storing commands into the Default and User Stores, you can protect individual commands from being changed through the use of the UNPROTECT command and password commands PRIVATE_PASSWORD and PUBLIC_PASSWORD. This feature is exclusive to Intersil devices.

Protecting individual commands is done by using both the PRIVATE_PASSWORD and UNPROTECT commands. First, the UNPROTECT command has a 32-byte long bit-vector in which each bit represents a PMBus command code. Setting the representative bit to 0 protects that command, meaning that command's value cannot be changed unless you attain a security level of 2 or 3, depending on whether the UNPROTECT string is stored in the User or Default Store, respectively. Attaining this security level of 2 or 3 is done by writing a password to the PRIVATE_PASSWORD command. The PRIVATE_PASSWORD is a 9-byte string, which by default is set to all null characters (0x000000000000000000). A PRIVATE_PASSWORD may be stored in the User Store as well as the Default Store, which means two levels of command protection. More information on these commands can be found in AN2013, and information on creating an UNPROTECT string can be found in Appendix D.

In addition to protecting individual commands, one may also protect all commands from being written by setting the PUBLIC_PASSWORD. The PUBLIC_PASSWORD is a 4-byte string, which by default is set to all null characters (0x00000000), and is only stored in the USER_STORE. When this value is set to all null characters, the device starts up in security level 1. This allows for commands to be written to unprotected commands.

When this value is set to something other than this, it will start up in security level 0. When in this level, no command can be written until the matching PUBLIC_PASSWORD is written to get into security level 1, or a matching PRIVATE_PASSWORD is written to get into security level 2 or 3.

There are a number of different combinations of the levels of protection offered. See Table 1 and the examples on the following pages to see what levels of protection are possible. Appendix E includes an example of write-protecting all commands from being written.

TABLE 1.

PUBLIC PASSWORD	PRIVATE PASSWORDS	DESCRIPTION
Not Set	None Set	This configuration is as shown in Figure 6. No password protection is offered.
Set	None Set	This configuration is described in Password Example 1. Basic protection against writing the wrong data is offered, but this protection can be easily defeated by writing NULL to PRIVATE_PASSWORD.
Set OR Not Set	Set only in User Store	This is not recommended, as writing NULL to PRIVATE_PASSWORD grants default-level access.
Not Set	Set only in Default Store	This configuration, when used with an UNPROTECT string, provides basic protection against changing commands in the default store. See Password Example 2.
Set	Set only in Default Store	This configuration, when used with an UNPROTECT string, provides basic protection against changing commands in the default store. Protection from writing accidental data is offered. See Password Example 3
Not Set	Set in Default & User Store	This configuration, when used with UNPROTECT strings, provides two levels of protection against changing commands in the User and Default stores. See Password Example 4.
Set	Set in Default & User Store	This configuration, when used with an UNPROTECT string, provides basic protection against changing commands in the default store. Protection from writing accidental data is offered. See Password Example 5

Password Example 1

ONE PUBLIC PASSWORD, NO PRIVATE PASSWORDS

As described previously, this configuration provides basic protection against accidentally writing the wrong value. When the device is powered up, it will start in security level 0, preventing write access to any command until the matching PUBLIC_PASSWORD is written, or a null value to PRIVATE_PASSWORD is written. This level of security is adequate only for systems where there is no worry of outside bus access to the device, and the PUBLIC_PASSWORD is intended merely as a protection from writing commands. Figure 11 demonstrates a configuration file with a PUBLIC_PASSWORD. Figures 11 and 12 can be used to clear the Public Password afterwards. A power-cycle of the device should be performed after loading any of these configuration files.

```
# Perform actions for Default Store

RESTORE_FACTORY      # Clear
STORE_DEFAULT_ALL    # Default Store
STORE_USER_ALL       # and User Store

RESTORE_DEFAULT_ALL # Prepare Default
                    # for adding cmds

 # Insert configuration data
 # you want in Default Store

MFR_SERIAL           SSSNNN
MFR_ID               Example OEM
VOUT_MAX             5.0 #Volts
VOUT_COMMAND         3.3 #Volts

STORE_DEFAULT_ALL    # Store Settings

# Perform actions for User Store

RESTORE_USER_ALL # Prepare User
                 # for adding cmds

 # Insert configuration data
 # you want in User Store

VOUT_COMMAND         2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

#Write desired Public Password
#This puts us at security level 0
PUBLIC_PASSWORD      MyPW

#Write our null private password to
#get into security level 2 or 3
PRIVATE_PASSWORD 0x0000000000000000

STORE_USER_ALL       # Store Settings
```

FIGURE 10. STORING VALUES IN THE DEFAULT AND USER STORES, WITH PUBLIC PASSWORD PROTECTION

```
# Write Matching Public Password
# to get security level 1
PUBLIC_PASSWORD     MyPW

# Prepare for change to User Store
RESTORE_USER_ALL
# Write Matching Public Password
# again, as restore made Sec. Level 0
PUBLIC_PASSWORD     MyPW

#Clear Public Password
#This puts us at security level 0
PUBLIC_PASSWORD     0x00000000

#Write our null private password to
#get into security level 2 or 3
PRIVATE_PASSWORD 0x000000000000000000

STORE_USER_ALL      # Store Settings
```

FIGURE 11. CLEAR PUBLIC PASSWORD USING PREVIOUS PUBLIC PASSWORD AND NULL PRIVATE PASSWORD

```
#Get into security level 3
PRIVATE_PASSWORD 0x000000000000000000

# Prepare for change to User Store
RESTORE_USER_ALL
#Get into sec. level 3 again, as
# as restore made Sec. Level 0
PRIVATE_PASSWORD 0x000000000000000000

#Clear Public Password
#This puts us at security level 0
PUBLIC_PASSWORD     0x00000000

#Write our null private password to
#get into security level 2 or 3
PRIVATE_PASSWORD 0x000000000000000000
STORE_USER_ALL      # Store Settings
```

FIGURE 12. CLEAR PUBLIC PASSWORD USING ONLY A NULL PRIVATE PASSWORD

Figure 13 goes one step further and clears the User and Default stores.

As seen in Figures 10 thru 13, the PRIVATE_PASSWORD command is used before storing a changed PUBLIC_PASSWORD, even though the intention of the files is to leave PRIVATE_PASSWORD blank and only set or clear PUBLIC_PASSWORD. This is because anytime the PUBLIC_PASSWORD is written a value other than what it currently has stored, it will change the security level to 0. This even occurs if the currently stored value is null (0x00000000), or if the device is in a security level greater than public (e.g. security levels 2 and 3). Because of this, a PRIVATE_PASSWORD must be issued after the changed PUBLIC_PASSWORD is issued in order to store the change.

```
# Write Matching Public Password
# to get security level 1
PUBLIC_PASSWORD     MyPW

# Restore to factory settings
# This also sets the PUBLIC_PASSWORD back to null
# It will also put our security level to 0
RESTORE_FACTORY

#Write our null private password to
#get into security level 2 or 3
PRIVATE_PASSWORD 0x000000000000000000

STORE_DEFAULT_ALL   # Store factory settings
STORE_USER_ALL      # to Default & User stores
```

FIGURE 13. STORING VALUES IN THE DEFAULT AND USER STORES, WITH PUBLIC PASSWORD PROTECTION

Password Example 2

NO PUBLIC PASSWORD, ONE PRIVATE PASSWORD IN THE DEFAULT STORE

As described before, this configuration provides protection against individual commands being rewritten when used with a properly configured UNPROTECT string. The device will start in security level 1, and will allow for write access to any command in the User Store, except for those already protected in the Default Store. Additionally, one will need the stored Private Password to easily revert the User Store to factory defaults. Figure 14 demonstrates loading a configuration file with this level of password protection. Figures 15 and 16 demonstrate how to clear the password protection of the default store, and how to clear the entire part after this configuration has been loaded.

```
# Perform actions for Default Store

RESTORE_FACTORY     # Clear
STORE_DEFAULT_ALL   # Default Store
STORE_USER_ALL      # and User Store

RESTORE_DEFAULT_ALL # Prepare Default
                    # for adding cmds

# Insert configuration data you want in Default Store

MFR_SERIAL          SSSNNN
MFR_ID              Example OEM
VOUT_MAX            5.0 #Volts
VOUT_COMMAND        3.3 #Volts

# Protect the following commands for the default store:
# MFR_ID and VOUT_MAX, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT    0xFFFFDFDFEFFFFFFFFFFFFFFFFFFFFFFFFFFFDFFFFFFFFFFFFFFFFFFFEFFF

#Set desired password for default store
PRIVATE_PASSWORD    ExampleDP

STORE_DEFAULT_ALL   # Store Settings

# Perform actions for User Store

RESTORE_USER_ALL # Prepare User
                 # for adding cmds

 # Insert configuration data
 # you want in User Store

VOUT_COMMAND        2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

STORE_USER_ALL      # Store Settings
```

FIGURE 14. STORING VALUES IN THE DEFAULT AND USER STORES WITH PASSWORD PROTECTED VALUES IN THE DEFAULT STORE

In this example, the commands MFR_ID and VOUT_MAX cannot be changed in active memory or in the User Store. However, VOUT_COMMAND and MFR_SERIAL can still be changed as it was not protected via the UNPROTECT string.

```
#Gain security level 3 access, in case a Public Password
#or User-level Private Password Exists
PRIVATE_PASSWORD    ExampleDP

RESTORE_DEFAULT_ALL

#Gain security level 3 access
PRIVATE_PASSWORD    ExampleDP

#Reset Password
PRIVATE_PASSWORD    0x00000000000000000

#Reset Unprotect
UNPROTECT     0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

STORE_DEFAULT_ALL

#Perform a Soft Reset
#WARNING: This will not work if a public password
#         or user-level private password exists.
#         You will need to power-cycle the device.
RESTORE_FACTORY
RESTORE_DEFAULT_ALL
RESTORE_USER_ALL
```

FIGURE 15. HOW TO CLEAR THE DEFAULT PASSWORD AND COMMAND PROTECTION AFTER THE CONFIGURATION FILE FROM FIGURE 14 IS LOADED

Please note that this only makes changes to the Default store.

```
#Gain security level 3 access
PRIVATE_PASSWORD    ExampleDP

#Restore to factory settings, which will reset
#command protection and password when stored.
#Security level reverts to either 1 or 0 when written,
#depending on presence of a public password.
RESTORE_FACTORY

#Gain security level 3 access again
PRIVATE_PASSWORD    ExampleDP

#Reset Password
PRIVATE_PASSWORD    0x00000000000000000

STORE_DEFAULT_ALL   #clear default store
                    #this will make security level 0 if using a PUBLIC_PASSWORD

#This next command is only required when using
#a PUBLIC_PASSWORD to restore the
#security level
PRIVATE_PASSWORD    0x00000000000000000

STORE_USER_ALL      #clear user store

#Perform a Soft Reset
RESTORE_FACTORY
RESTORE_DEFAULT_ALL
RESTORE_USER_ALL
```

FIGURE 16. HOW TO CLEAR THE ENTIRE DEVICE MEMORY AFTER THE CONFIGURATION FILE FROM FIGURE 14 IS LOADED

Password Example 3

ONE PUBLIC PASSWORD, ONE PRIVATE PASSWORD IN THE DEFAULT STORE

This example adds on to Password Example 2 with a Public Password. This configuration provides basic protection against accidentally writing the wrong value. When the device is powered up, it will start in security level 0, preventing write access to any command until the matching PUBLIC_PASSWORD is written to gain security level 1, or a null value to PRIVATE_PASSWORD is written to gain security level 2. See Figure 17 for the example configuration. Refer back to Figure 15 to clear the Private Password, Figure 11 to clear the Public Password, and Figure 16 to clear the device memory after the configuration in Figure 17 has been loaded.

```
# Perform actions for Default Store

RESTORE_FACTORY     # Clear
STORE_DEFAULT_ALL   # Default Store
STORE_USER_ALL      # and User Store

RESTORE_DEFAULT_ALL # Prepare Default Store for adding cmds

# Insert configuration data you want in Default Store
MFR_SERIAL          SSSNNN
MFR_ID              Example OEM
VOUT_MAX            5.0 #Volts
VOUT_COMMAND        3.3 #Volts

# Protect the following commands for the default store:
# MFR_ID and VOUT_MAX, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT     0xFFFFFDFFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDFFFFFFFFFFFFFFFFFFFF

#Set desired password for default store
PRIVATE_PASSWORD    ExampleDP
STORE_DEFAULT_ALL   # Store Settings


# Perform actions for User Store

RESTORE_USER_ALL # Prepare User Store for adding cmds

# Insert configuration data you want in User Store
VOUT_COMMAND        2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

#Set Public Password (which will bring security level to 0)
PUBLIC_PASSWORD     MyPW

#Bring security level to 2, in order to store settings
PRIVATE_PASSWORD    0x00000000000000000

STORE_USER_ALL      # Store Settings

RESTORE_USER_ALL    # This restore is performed to let the password
                    # settings / security take effect. Power-cycling
                    # the device would have the same effect.
```

FIGURE 17. STORING VALUES IN THE DEFAULT AND USER STORES WITH PASSWORD PROTECTED VALUES IN THE DEFAULT STORE, AND A PUBLIC PASSWORD PREVENTING WRITES ON START-UP

In this example, the commands MFR_ID and VOUT_MAX cannot be changed in active memory or in the User Store. However, VOUT_COMMAND and MFR_SERIAL can still be changed as it was not protected via the UNPROTECT string.

Password Example 4

NO PUBLIC PASSWORD, TWO PRIVATE PASSWORDS IN USER/DEFAULT STORE

This example adds on to Password Example 2 with a second Private Password in the User Store. When the device is powered up, it will start in security level 1, and will prevent write access to commands protected in either the user or default store as dictated by their UNPROTECT strings. See Figure 18 for this configuration. See Figure 19 (which is the same as Figure 15) to clear the Default Store password and its command protection, Figure 20 to clear the Default Store password / command protection, and refer back to Figure 16 to clear the entire device memory.

```
# Perform actions for Default Store
RESTORE_FACTORY     # Clear
STORE_DEFAULT_ALL   # Default Store
STORE_USER_ALL      # and User Store

RESTORE_DEFAULT_ALL # Prepare Default Store for adding cmds

# Insert configuration data you want in Default Store
MFR_SERIAL          SSSNNN
MFR_ID              Example OEM
VOUT_MAX            5.0 #Volts
VOUT_COMMAND        3.3 #Volts

# Protect the following commands for the default store:
# MFR_ID and VOUT_MAX, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT    0xFFFFFDFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDFFFFFFFFFFFFFFFFFFFEFFF

# Set desired password for default store
PRIVATE_PASSWORD    ExampleDP
STORE_DEFAULT_ALL   # Store Settings

# Perform actions for User Store
RESTORE_USER_ALL # Prepare User Store for adding cmds

# Insert configuration data you want in User Store
VOUT_COMMAND        2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

# Protect the following command for the User store: VOUT_COMMAND, as well as
# RESTORE_FACTORY, STORE_DEFAULT_ALL, RESTORE_DEFAULT_ALL, STORE_USER_ALL.
# The last four are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT  0xFFFFD9FFFDFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFF

#Bring security level to 2, in order to store settings
PRIVATE_PASSWORD    ExampleUP
STORE_USER_ALL      # Store Settings

RESTORE_USER_ALL    # This restore is performed to let the password
                    # settings / security take effect. Power-cycling
                    # the device would have the same effect.
```

FIGURE 18. STORING VALUES IN BOTH THE DEFAULT STORE AND USER STORE WITH PASSWORD PROTECTION

In this example, the commands MFR_ID and VOUT_MAX cannot be changed in active memory or in the User Store. Additionally, VOUT_COMMAND cannot be changed in active memory, as it is protected in the User Store. However, PID_TAPS and MFR_SERIAL can still be changed as it was not protected via the User or Default Store UNPROTECT string.

```
#Gain security level 3 access, in case a Public Password
#or User-level Private Password Exists
PRIVATE_PASSWORD    ExampleDP

RESTORE_DEFAULT_ALL #Prepare for command modification to default store

#Gain security level 3 access
PRIVATE_PASSWORD    ExampleDP

#Reset Password
PRIVATE_PASSWORD    0x000000000000000000

#Reset Unprotect
UNPROTECT    0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

STORE_DEFAULT_ALL

#Please power-cycle the device after loading this file
```

FIGURE 19. HOW TO CLEAR THE DEFAULT-LEVEL PRIVATE PASSWORD AND COMMAND PROTECTION AFTER THE CONFIGURATION FILE FROM FIGURE 18 OR 21 IS LOADED

Please note that the file is similar in operation as Figure 15.

```
#Gain security level 2 access, in case a
#Public Password Exists
PRIVATE_PASSWORD    ExampleUP

RESTORE_USER_ALL #Prepare for command modification to default store

#Gain security level 2 access
PRIVATE_PASSWORD    ExampleUP

#Reset Password
PRIVATE_PASSWORD    0x000000000000000000

#Reset Unprotect
UNPROTECT    0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

STORE_USER_ALL

RESTORE_USER_ALL    # This restore is performed to let the password
                    # settings / security take effect. Power-cycling
                    # the device would have the same effect.
```

FIGURE 20. HOW TO CLEAR THE USER-LEVEL PRIVATE PASSWORD AND COMMAND PROTECTION AFTER THE CONFIGURATION FILE FROM FIGURE 18 OR 21 IS LOADED

Password Example 5

ONE PUBLIC PASSWORD, TWO PRIVATE PASSWORDS IN USER/DEFAULT STORE

This example adds on to Password Example 4 with a Public Password. When the device is powered up, it will start in security level 0, and will prevent write access to any commands. Once the public password is entered, we will be in security level 1, and will prevent write access to commands protected in either the User or Default store as dictated by their UNPROTECT strings. See Figure 21 for this configuration. Refer back to Figure 18 to clear the Private Password and command protection in the default store, and Figure 19 to clear the Private Password and command protection in the user store. Refer to Figure 22 to clear the Public Password. Figure 15 may again be used to clear both the User and Default Stores.

```
# Perform actions for Default Store

RESTORE_FACTORY     # Clear
STORE_DEFAULT_ALL   # Default Store
STORE_USER_ALL      # and User Store

RESTORE_DEFAULT_ALL # Prepare Default Store for adding cmds

# Insert configuration data you want in Default Store
MFR_SERIAL          SSSNNN
MFR_ID              Example OEM
VOUT_MAX            5.0 #Volts
VOUT_COMMAND        3.3 #Volts

# Protect the following commands for the default store:
# MFR_ID and VOUT_MAX, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT    0xFFFFFDFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDFFFFFFFFFFFFFFFFFFFEFFF

# Set desired password for default store
PRIVATE_PASSWORD    ExampleDP
STORE_DEFAULT_ALL   # Store Settings


# Perform actions for User Store

RESTORE_USER_ALL # Prepare User Store for adding cmds

# Insert configuration data you want in User Store
VOUT_COMMAND        2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

# Protect the following command for the User store: VOUT_COMMAND, as well as
# RESTORE_FACTORY, STORE_DEFAULT_ALL, RESTORE_DEFAULT_ALL, STORE_USER_ALL.
# The last four are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT  0xFFFFD9FFFDFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFF

#Set Public Password, which will change security level to 0
PUBLIC_PASSWORD     MyPW

#write null private password to get security level to 2
PRIVATE_PASSWORD    0x000000000000000000

#set private password
PRIVATE_PASSWORD    ExampleUP

STORE_USER_ALL      # Store Settings

RESTORE_USER_ALL    # This restore is performed to let the password
                    # settings / security take effect. Power-cycling
                    # the device would have the same effect.
```

FIGURE 21. STORING VALUES IN THE DEFAULT AND USER STORES WITH PASSWORD PROTECTED VALUES IN THE DEFAULT STORE AND USER STORE, WITH THE ADDITION OF PUBLIC PASSWORD PROTECTION

In this example, the commands MFR_ID and VOUT_MAX cannot be changed in active memory or in the User Store. Additionally, VOUT_COMMAND cannot be changed in active memory, as it is protected in the User Store. However, PID_TAPS and MFR_SERIAL can still be changed as neither commands were protected via the User or Default Store UNPROTECT string. Please note that any attempted write after loading this file can only be done after a matching PUBLIC_PASSWORD or PRIVATE_PASSWORD is written.

```
# Write Matching Public Password
# to get security level 1
PUBLIC_PASSWORD     MyPW

# Prepare for change to User Store
RESTORE_USER_ALL
# Write Matching Public Password
# again, as restore made Sec. Level 0
PUBLIC_PASSWORD     MyPW

#Clear Public Password
#This puts us at security level 0
PUBLIC_PASSWORD     0x00000000

#Get into Security Level 2 to store
#changes
PRIVATE_PASSWORD    ExampleUP

STORE_USER_ALL      # Store Settings

RESTORE_USER_ALL    # This restore is performed to let the password
                    # settings / security take effect. Power-cycling
                    # the device would have the same effect.
```

FIGURE 22. CLEAR PUBLIC PASSWORD USING PREVIOUS PUBLIC PASSWORD AND USER STORE PRIVATE PASSWORD

Appendix D: Generating Unprotect Strings

In order to take advantage of protecting individual commands via PRIVATE_PASSWORD, the value of the UNPROTECT command to make set appropriately. The UNPROTECT command has a 32-byte long bit-vector in which each bit represents a PMBus command code. Setting the representative bit to 0 protects that command, meaning that command's value cannot be changed unless you attain a security level of 2 or 3 - depending on whether the UNPROTECT command is stored in the User or Default Store, respectively.

To make the task of creating the UNPROTECT command value easier, Intersil provides an UNPROTECT string generation tool in all of its command spreadsheets. These spreadsheets are device-dependent, and are included in the AN2031 examples and files attachment. To use a command spreadsheet to generate an UNPROTECT value, read the following steps:

Step 1: Open the command spreadsheet appropriate for the device being used. Upon opening, you will notice a number of different columns such as "PMBus Command", "Command Code", and most importantly, the "Protect?" column, as seen in Figure 23. You will learn how to set the values in the Protect column to generate an UNPROTECT string.

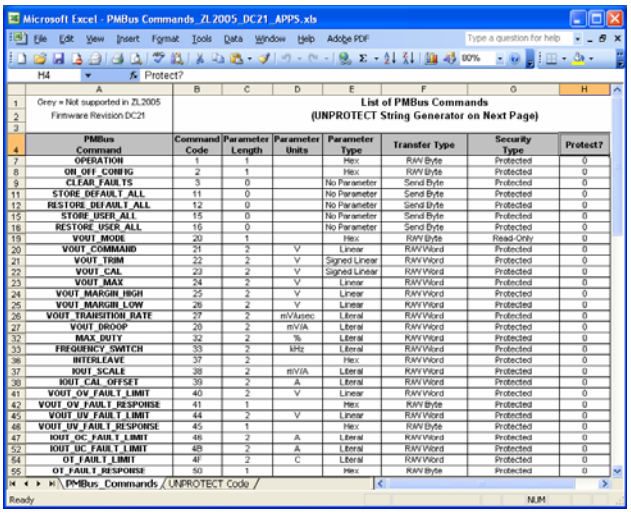
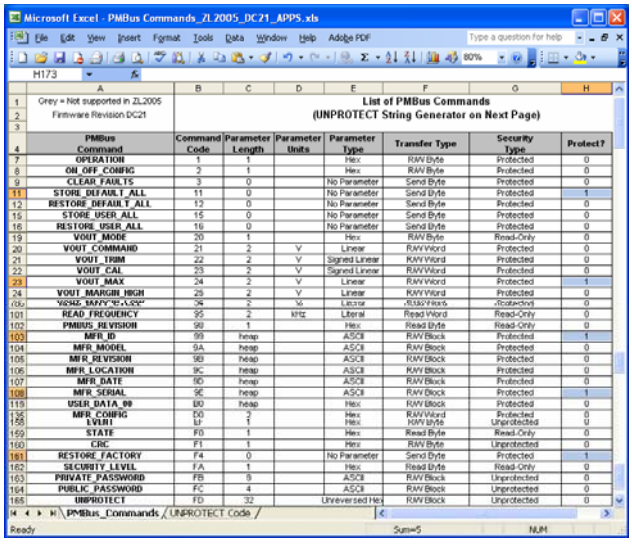


FIGURE 23. A ZL2005 PMBus COMMAND SPREADSHEET

Step 2: As an example, let's create an unprotect string similar to Password Example 2, but this time also protect the MFR_SERIAL command. Overall, the commands we want to protect are MFR_SERIAL, MFR_ID, and VOUT_MAX.

Additionally, the commands RESTORE_FACTORY and STORE_DEFAULT_ALL must be protected when creating an UNPROTECT string that is to be stored in the Default Store. This is needed to ensure that there will be no backdoor to overwrite data via these commands. If storing to the User Store, the commands RESTORE_FACTORY, STORE_DEFAULT_ALL, RESTORE_DEFAULT_ALL, and STORE_USER_ALL must all be protected for command protection to work.

To select the above commands to be protected in the spreadsheet, find the "Protect?" column that corresponds to the command you want to protect. Then change the value from "0" to "1", as shown in Figure 24. This change causes the unprotect string on the "UNPROTECT Code" page to be recalculated, as you will see in Step 3. Note that the bit inversion for the command's bit-vector is done automatically.



Step 3: Now that the commands to be protected are set, go to the "UNPROTECT Code" page. The sixth row on this page contains the UNPROTECT string. This cell can be copied and pasted as a new line in a configuration file. It is advised that you make sure that the UNPROTECT value does not appear as all "FF" values, unless you are trying to remove command protection. On some installations of Excel, an extra toolpack may need to be installed in order for this calculation to work.

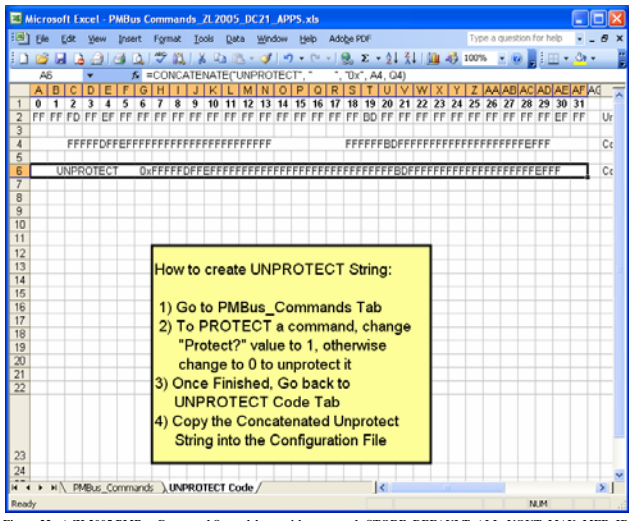


FIGURE 24. A ZL2005 PMBus COMMAND SPREADSHEET, WITH COMMANDS STORE_DEFAULT_ALL, VOUT_MAX, MFR_ID, MFR_SERIAL, and RESTORE_FACTORY SELECTED FOR PROTECTION

Step 4: Copy and paste the UNPROTECT string into the configuration file. Following the similar structure from Example 2, the configuration file should look similar to as shown in Figure 25.

```
# Perform actions for Default Store

RESTORE_FACTORY      # Clear
STORE_DEFAULT_ALL    # Default Store
STORE_USER_ALL       # and User Store

RESTORE_DEFAULT_ALL # Prepare Default
                    # for adding cmds

# Insert configuration data you want in Default Store

MFR_SERIAL           SSSNNN
MFR_ID               Example OEM
VOUT_MAX             5.0 #Volts
VOUT_COMMAND         3.3 #Volts

# Protect the following commands for the default store:
# MFR_SERIAL, MFR_ID, and VOUT_MAX, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT     0xFFFFFDFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBDFFFFFFFFFFFFFFFFFFFFEFF

#Set desired password for default store
PRIVATE_PASSWORD     ExampleDP

STORE_DEFAULT_ALL    # Store Settings

# Perform actions for User Store

RESTORE_USER_ALL # Prepare User
                 # for adding cmds

 # Insert configuration data
 # you want in User Store

VOUT_COMMAND         2.5 #Volts
PID_TAPS  A=1634, B=-2799, C=1227

STORE_USER_ALL       # Store Settings
```

FIGURE 25. STORING VALUES IN THE DEFAULT AND USER STORES WITH PASSWORD PROTECTED VALUES IN THE DEFAULT STORE

```
# Configuration file for ZL2005-002-DC21  at Device Address 0x20 MODIFIED
#------------------------------------------------------------
# Clear Memory
# WARNING: Make sure no Password Protection is set before
#          loading this config file.
#------------------------------------------------------------
RESTORE_FACTORY
STORE_DEFAULT_ALL
STORE_USER_ALL

#------------------------------------------------------------
# Default Store Data
#------------------------------------------------------------
# The next line is required to insert Default Store parameters
RESTORE_DEFAULT_ALL
#VOUT_COMMAND                 3.300049
VOUT_COMMAND                 0x699A
#VOUT_MAX                     5.000000
VOUT_MAX                     0xA000

MFR_ID                       Example OEM
MFR_SERIAL                   SSSNNN

# Protect MFR_ID, MFR_SERIAL and VOUT_MAX, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
UNPROTECT  0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

# Set desired password for default store
PRIVATE_PASSWORD     ExampleDP

STORE_DEFAULT_ALL  #uncomment to store above settings

#------------------------------------------------------------
# Current Settings/User Store Data
#------------------------------------------------------------
# The next line is required to insert User Store parameters
RESTORE_USER_ALL
#VOUT_COMMAND                 2.500000
VOUT_COMMAND                 0x5000
#PID_TAPS                     A=1634.000000, B=-2799.000000, C=1227.000000
PID_TAPS                     A=0x7BCC40, B=0xFCAEF0, C=0x7B9960

# Protect VOUT_COMMAND, as well as
# RESTORE_FACTORY, STORE_DEFAULT_ALL, RESTORE_DEFAULT_ALL, STORE_USER_ALL.
UNPROTECT  0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

#Set Public Password, which will change security level to 0
PUBLIC_PASSWORD      MyPW
#write null private password to get security level to 2
PRIVATE_PASSWORD     0x00000000000000000
#set private password
PRIVATE_PASSWORD     ExampleUP

STORE_USER_ALL  #uncomment to store above settings

#------------------------------------------------------------
# Soft Reset of Device
#------------------------------------------------------------
#RESTORE_FACTORY
#RESTORE_DEFAULT_ALL
RESTORE_USER_ALL
```

In this example, the commands MFR_SERIAL, MFR_ID, and VOUT_MAX cannot be changed in active memory or in the User Store. However, VOUT_COMMAND can still be changed as it was not protected via the UNPROTECT string.

Appendix E: Blocking Write Access to All Commands

For some applications of Intersil device, a board designer may desire to prevent their product from accepting write commands over the digital bus. For example, a designer of a DC:DC power supply module may decide that the digital bus port of the module should be used for monitoring of the module, but not for programming or controlling the module.

The first way of providing such write protection can be done just by storing PUBLIC_PASSWORD as a non-null value in the User Store, as demonstrated in earlier examples. However, if more protection is desired from writing to ANY command, the following example (see Figure 26) stores a number of settings in the Default Store, but protects all commands from being modified, including commands that don't have values set, via the UNPROTECT string. This example also sets protection redundancies, in the form of setting a Private Password and Public Password in the User Store, as well as setting the UNPROTECT string in the User Store to prohibit any command writes.

The example shown in Figure 26 is similar in structure to Password Example 5. Because of this, Figure 16 may be used to clear the device memory after this file is loaded.

```
# Clear device memory - All passwords must be cleared before loading this file

RESTORE_FACTORY     # Clear
STORE_DEFAULT_ALL   # Default Store
STORE_USER_ALL      # and User Store

#Perform actions for Default Store

RESTORE_DEFAULT_ALL # Prepare Default Store for adding cmds

# Insert configuration data you want in Default Store
MFR_SERIAL          SSSNNN
MFR_ID              Example OEM
VOUT_MAX            5.0 #Volts
VOUT_COMMAND        3.3 #Volts

# Protect the following commands for the default store:
# ALL COMMANDS, as well as
# RESTORE_FACTORY and STORE_DEFAULT_ALL - which are required to ensure security.
# RESTORE_DEFAULT_ALL, RESTORE_USER_ALL, STORE_USER_ALL, and UNPROTECT are
# left unprotected to allow storing protection in the User Store
# NOTE: This UNPROTECT string is intended for a ZL2005
UNPROTECT  0x000064000000000000000000000000000000000000000000000000000000000038

# Set desired password for default store
PRIVATE_PASSWORD    ExampleDP
STORE_DEFAULT_ALL   # Store Settings


# Perform actions for User Store

RESTORE_USER_ALL # Prepare User Store for adding cmds

# NO COMMANDS CAN BE SET IN USER (except for UNPROTECT and Passwords)
# DUE TO UNPROTECT SETTINGS IN DEFAULT STORE

# Protect everything (set to all 0's)
UNPROTECT  0x00000000000000000000000000000000000000000000000000000000000000000000

#Set Public Password, which will change security level to 0
PUBLIC_PASSWORD     MyPW

#write null private password to get security level to 2
PRIVATE_PASSWORD    0x000000000000000000

#set private password
PRIVATE_PASSWORD    ExampleUP

STORE_USER_ALL      # Store Settings

RESTORE_USER_ALL    # This restore is performed to let the password
                    # settings / security take effect. Power-cycling
                    # the device would have the same effect.
```

FIGURE 26. A CONFIGURATION FILE GENERATED BY POWERNAVIGATOR, WITH SOME EXAMPLE PASSWORD PROTECTION INSERTED, UNPROTECT STRINGS SET, AND FOOTER COMMANDS SET APPROPRIATELY

References

AN2013 - PMBus Commands, Zilker Labs, 2007.

AN2026 - PowerNavigator User's Guide, Zilker Labs, 2008.

AN2033 - PMBus Commands for DDC Devices, Zilker Labs, 2008.

AN2035 - Compensating DDC Devices Using CompZL, Zilker Labs, 2008.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

© 2018 Renesas Electronics Corporation. All rights reserved.

Colophon 7.0