

## Getting Started with BelaSigna™ R261



**ON Semiconductor®**

<http://onsemi.com>

---

### APPLICATION NOTE

#### Introduction

This application note describes the procedure to start working with BelaSigna R261 hardware, configure the application per your specific needs, create a custom application and finally create an EEPROM binary file for your target hardware. It is intended for FAEs and customers working with BelaSigna R261 based applications.

More specifically, you will learn about the following items:

1. Hardware and software requirements
  - Description of the necessary hardware and software components that you will need to follow content of this application note
2. How to communicate with your device
  - Discover how to establish a communication between the SignaKlara Device Utility (SKDU) and your hardware, including some troubleshooting advice
3. Real-time configuration of your device
  - Learn how to use the BelaSigna R261 configuration tab BelaSigna R261 within the SKDU, and discover how you can use the I2C commands in addition to all functions built inside the SKDU to check the configuration of your BelaSigna R261
  - Learn how to modify these settings
  - Learn how to switch between different operating modes
  - Understand how to change the output stage configuration, etc.
4. How to use the diagnostic functions
  - Learn how to use the diagnostic capabilities of BelaSigna R261, for example, to calibrate the output volume with your recording hardware
5. How to configure the built-in algorithms
  - Learn how to change the BSE algorithm parameters, for example changing from Far-Talk to Close-Talk, as well as using the Custom Mode to download a specific version of the BSE algorithm, using a .d file, allowing, for example, to configure BelaSigna R261 to implement a Near-Talk mode
  - Learn the use of the built-in equalizer -- it will show you how you can implement a specific frequency shaping, using the equalizer
6. How to create custom applications
  - After experiencing the run-time configuration of BelaSigna R261, you will be guided through the use of the application template, also known as the .abst file. These sections will help you convert all your run-time configuration changes into a format that you can reuse in the future.
  - Learn how you can create a custom .abs file with all your desired modifications to the ROM based application
  - Discover the process of creating an EEPROM image with these settings.
7. Various tips
  - How to create an Intel HEX-format EEPROM file
  - How to create command-line scripts to download software to an SPI EEPROM attached to a BelaSigna R261 without using the SKDU
  - How to restore an empty EEPROM, if you want to ensure that BelaSigna R261 boots from its default ROM application

## Hardware and Software Components Required

It is assumed that the hardware is already working, and that you have all the necessary material to communicate with your BelaSigna R261. As a reminder, the following are required:

- A working hardware target with BelaSigna R261 (26 or 30-ball version)
- The I2C interface pins (I2C\_SDA, I2C\_SCL and a ground) available on test points
- A Communication Accelerator Adaptor (CAA) and the appropriate I2C cable
- The correct drivers for your CAA (can be downloaded from MyON)
- The SignaKlara Device Utility (SKDU), version 1.2 (also available from MyON)
- An external EEPROM attached to BelaSigna R261 if you wish to use a custom EEPROM image

If you wish to experiment the procedures and guidelines discussed in this application note, you need to obtain a software package that will contain all the necessary files and applications that are used and described in this document. It is a zip archive file that has been encrypted with the password ‘br261’ (without the quotes). Contact your ON Semiconductor technical support to get this package.

For detailed information on BelaSigna R261, please refer to the device datasheet, which can be found on [www.onsemi.com](http://www.onsemi.com). Other documentation is also available; please contact ON Semiconductor technical support for more information.

## Establish the Communication with Your Device

The first step in this procedure is to ensure your SKDU is properly communicating with your BelaSigna R261. Follow these steps to verify this:

- Run the SKDU (e.g., Start Menu → All Programs → ON Semiconductor → SignaKlara Device Utility)
- The SKDU can sometimes take quite a bit of time to automatically detect your hardware, so you may want to edit the “Auto Configuration” preference in the Windows → Preferences menu, such that it only searches on the interface(s) you are likely to use. See a configuration example in the picture below. Note that if you only intend to use the SKDU with BelaSigna R261, then the only interface you need to keep in this dialog is the I2C interface of the Communication Accelerator Adaptor. Click Apply and OK in this form, should you decide to update your auto configuration preferences.

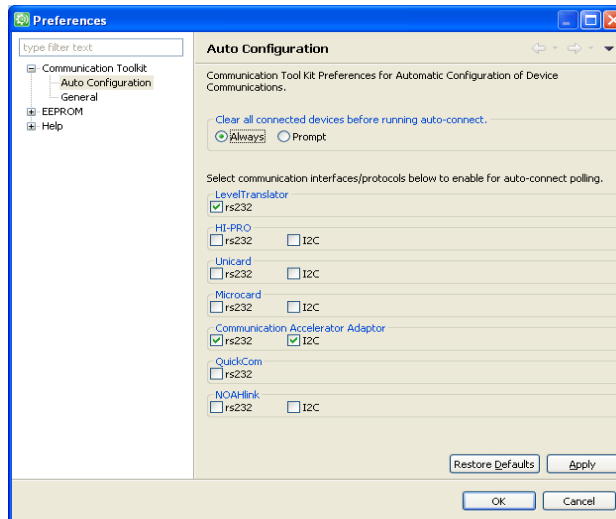
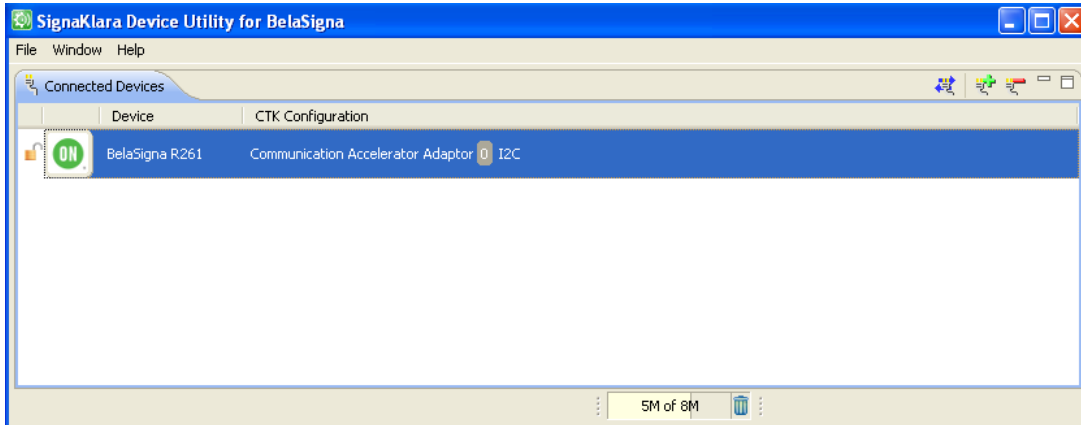


Figure 1. Change your Auto Configuration Preferences

## AND8485

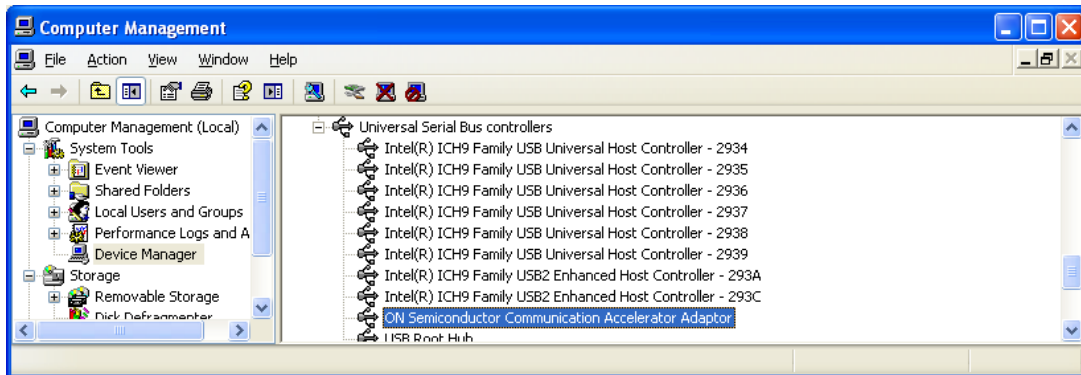
- If your BelaSigna R261 is properly running, the SKDU will appear as in the following figure, observe the little unlocked padlock icon on the left of the ON Semiconductor logo:



**Figure 2. Successful connection to BelaSigna R261**

If your SKDU does not appear as on Figure 2, a few things might need to be verified:

1. You do not see the BelaSigna R261 device in the “Connected Devices” view of the SKDU
  - Check that your CAA is properly connected to your computer’s USB port, and that its main power switch is turned on. This can be confirmed by checking the LED that is next to the CAA power switch.
  - Check that your CAA driver is properly installed. You should have a device called “ON Semiconductor Communication Accelerator Adaptor” in the USB Controller section of your computer’s device manager. If you do not have this driver installed, download the latest from the MyON website, and re-install it.

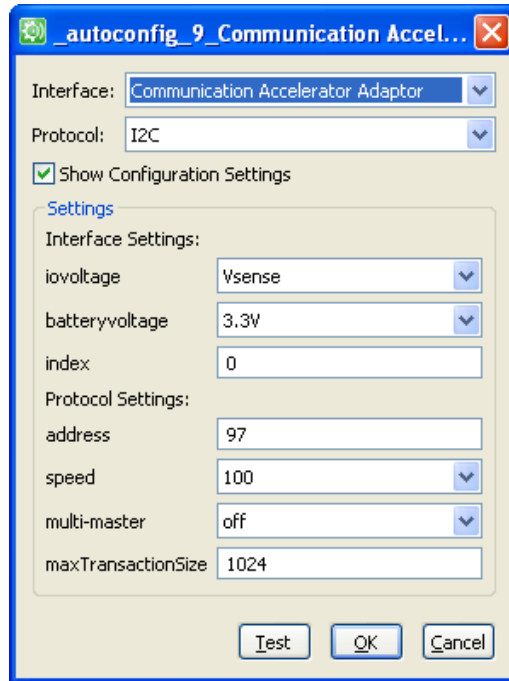


**Figure 3. CAA Device Driver**

- Verify that the I2C cable is properly connected between your CAA and your BelaSigna R261, you need three connections to ensure working communication:
  - The I2C clock signal (I2C\_SCL)
  - The I2C data line (I2C\_SDA)
  - A ground connection.
- Check that your device is properly powered by monitoring the VBAT pin, and ensuring that the voltage on this pin is within the supported power supply range of BelaSigna R261, usually between 1.8 V and 3.3 V.
- You also need to ensure that BelaSigna R261 is receiving the proper clock frequency. Check the clock frequency of the EXT\_CLK signal on BelaSigna R261. Then, measure the voltage level on the CONFIG\_SEL pin. It is usually defined by a resistive divider, which configures BelaSigna R261 in a specific clocking mode. The clock frequency applied to EXT\_CLK pin should be within 10% of the clock configuration defined by CONFIG\_SEL. If you are using the Customer Prototype Platform (CPP) with its onboard clock, then the frequency is 2.048 MHz, and you need to set the CONFIG\_SEL octal switch on position 0 to ensure proper operation. Check the device datasheet for additional information on the hardware configuration of BelaSigna R261.

## AND8485

2. You can see the BelaSigna R261 device in your SKDU, but there is a yellow question mark on top of the unlocked padlock icon.
  - Check the voltage levels: voltage levels need to be properly defined to ensure proper communication with BelaSigna R261. The power supply applied on BelaSigna R261’s VDDO pin needs to be matched with the voltage levels defined in the SKDU. To verify the voltage levels used by the SKDU, right-click on your BelaSigna device in the “Connected Device” view of the SKDU, and select “Edit Config”. Ensure that the right voltage levels are defined. An example can be seen on Figure 4:



**Figure 4. CTK Configuration Example**

The only parameters that you might need to change in the above dialog are the “iovoltage” and “batteryvoltage” parameters; using “V<sub>sense</sub>” for “iovoltage” tells the CAA to read the IO voltages from your target hardware. If you are intending to power your device with the CAA, you need to ensure that the device doesn’t consume too much current. Typically, the CPP cannot be powered with the 1.8 V option by the CAA, so in this case, ensure you are using 3.3 V, as shown on Figure 4. Once you have set these two parameters, double click on your BelaSigna R261 device to verify proper communication. If your SKDU can display the following dialog, you are connected!

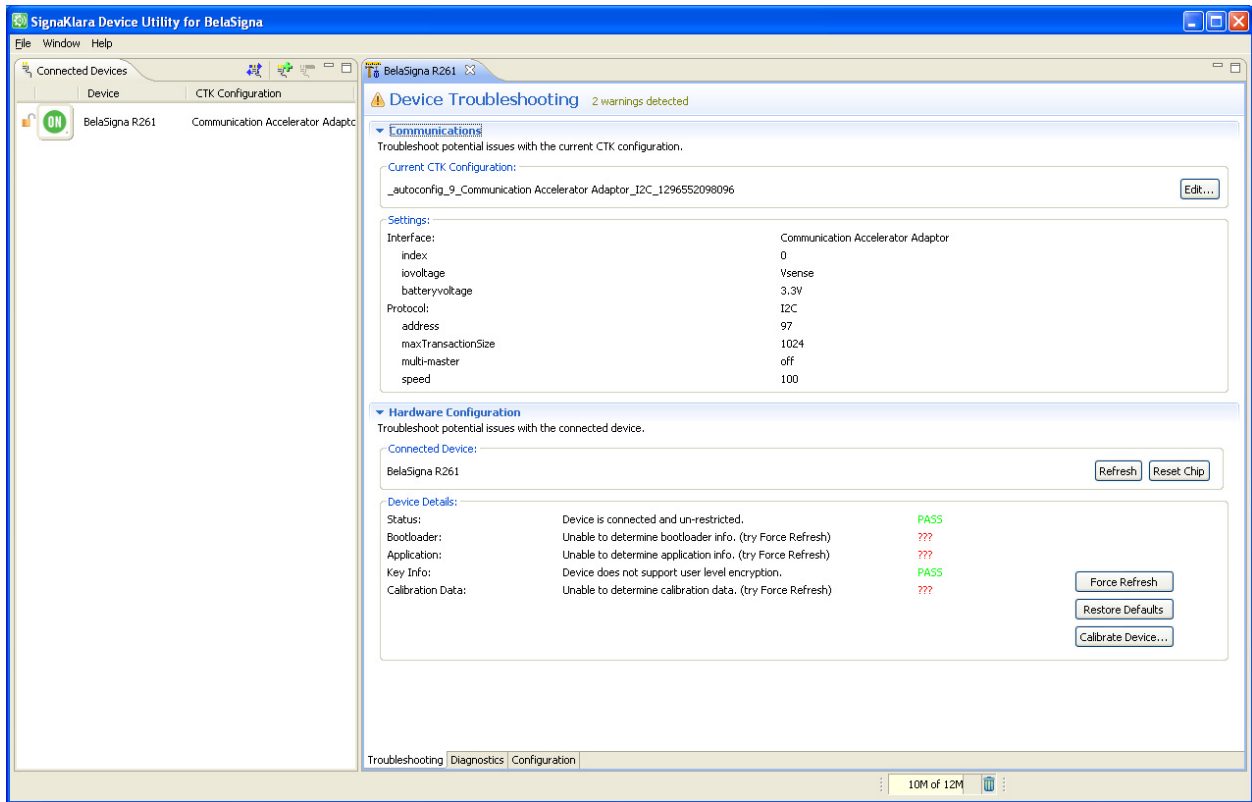


Figure 5. Device Details when Connected

You can verify the voltage levels that the SKDU automatically selected, or that you have manually configured previously. You can also observe a few “???” reports concerning the EEPROM content. In fact, BelaSigna R261 does not allow the SKDU to retrieve all this information without erasing some important areas of the application running inside BelaSigna R261. Consequently, the SKDU will not read these parameters, and will display the “???” marks. If you really want to check on the status of the EEPROM, you can hit the “Force Refresh” button, which will read the contents of your EEPROM. Note that the SKDU will modify the RAM contents to support this operation, but not the EEPROM contents. After such an operation, BelaSigna R261 needs to be rebooted to restart processing the noise reduction algorithm again. This can be done by hitting the “Restore Defaults” button in the SKDU.

If for some reason you see all PASS in the device details after running the “Force Refresh” button, as shown in Figure 6, it means that your EEPROM was already programmed, and to continue the process of this application note, you should ensure you modify the content of your EEPROM, such as your BelaSigna R261 does not boot from it. The last section of this application note describes how to perform this task.

# AND8485

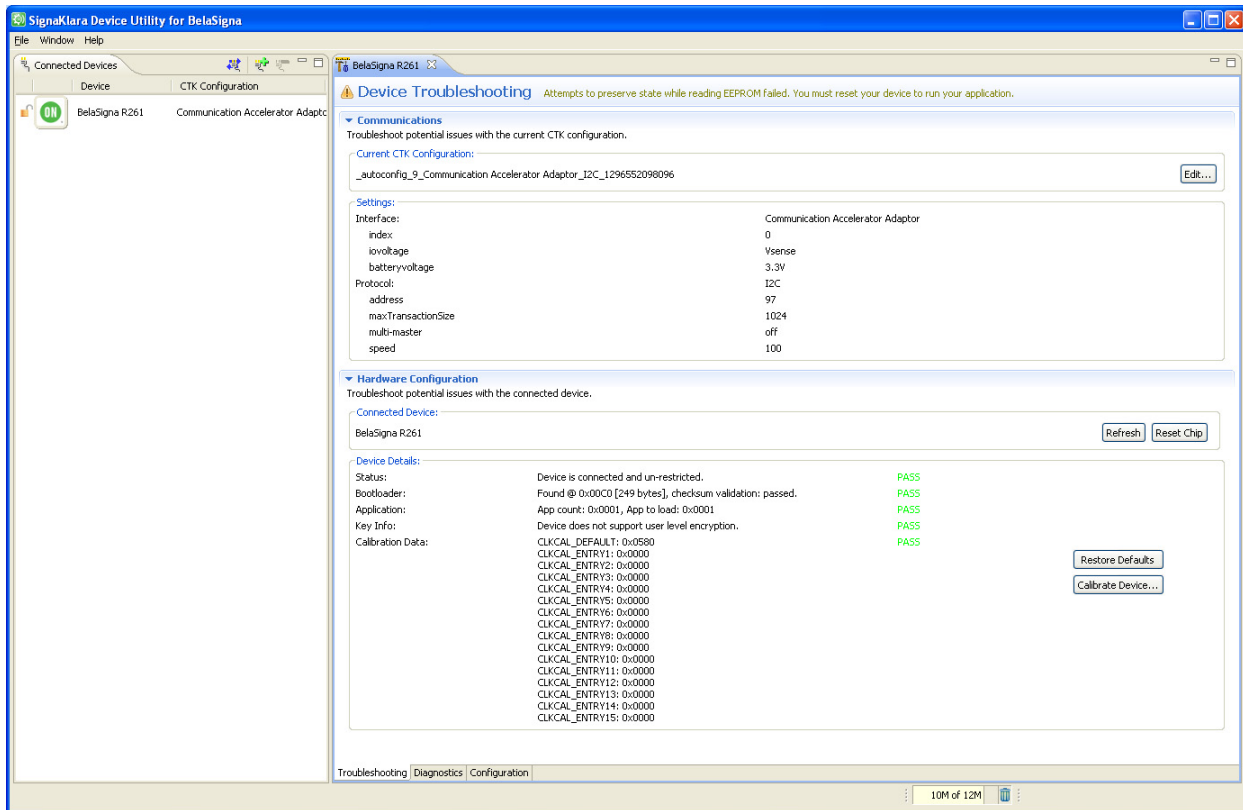


Figure 6. Device Details after "Force Refresh"

# AND8485

If your target hardware doesn't contain an EEPROM, or that you are using this hardware for the first time with an EEPROM that is very likely empty, you will get the following response from the SKDU when pressing the "Force Refresh" button:

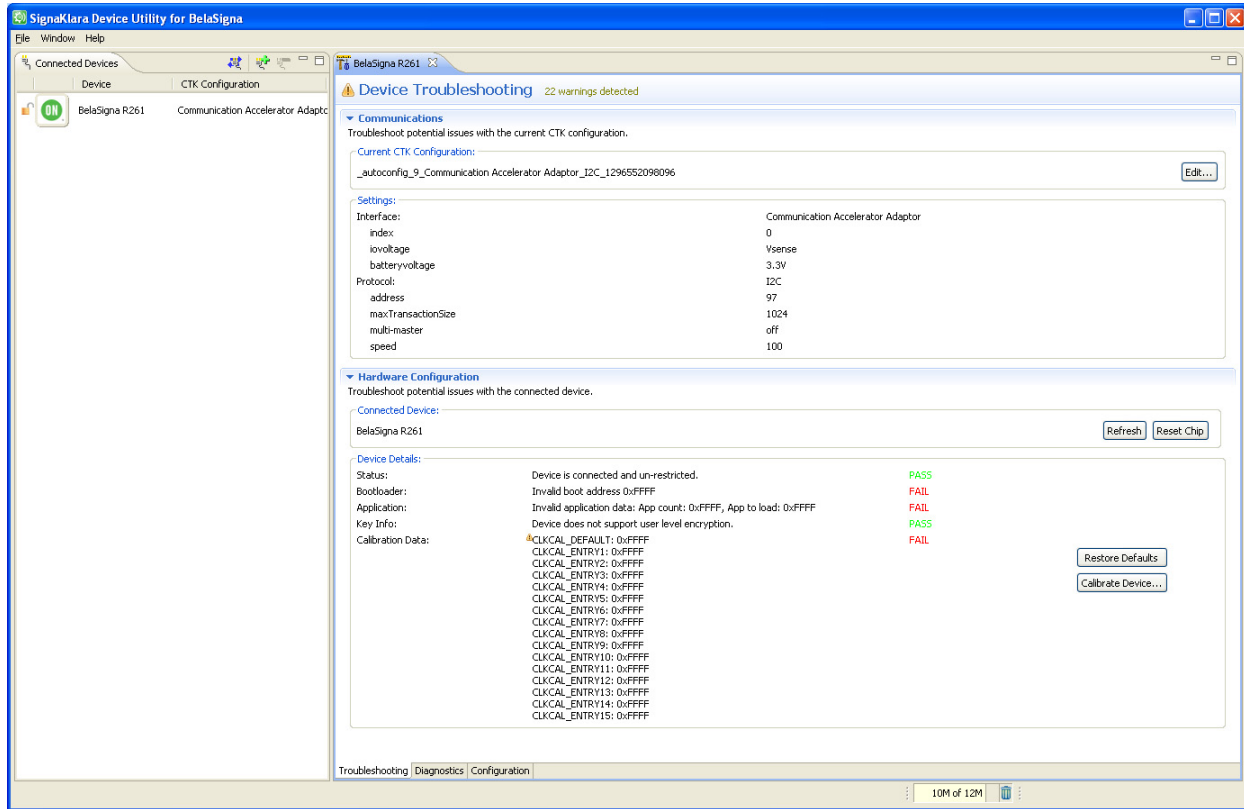


Figure 7. Device Details with no EEPROM (or empty)

# AND8485

## Check the Current Configuration, and Understand Interaction with BelaSigna R261 Within the SKDU

After you have established connection with your BelaSigna R261, click on the configuration tab (bottom of the BelaSigna R261 view). You will see the current configuration of your device. A pull-down menu is available, offering a number of I2C configuration commands. As an example, use the “Get\_Application\_Status” command to get information about the application currently running on your device, as per the following picture:

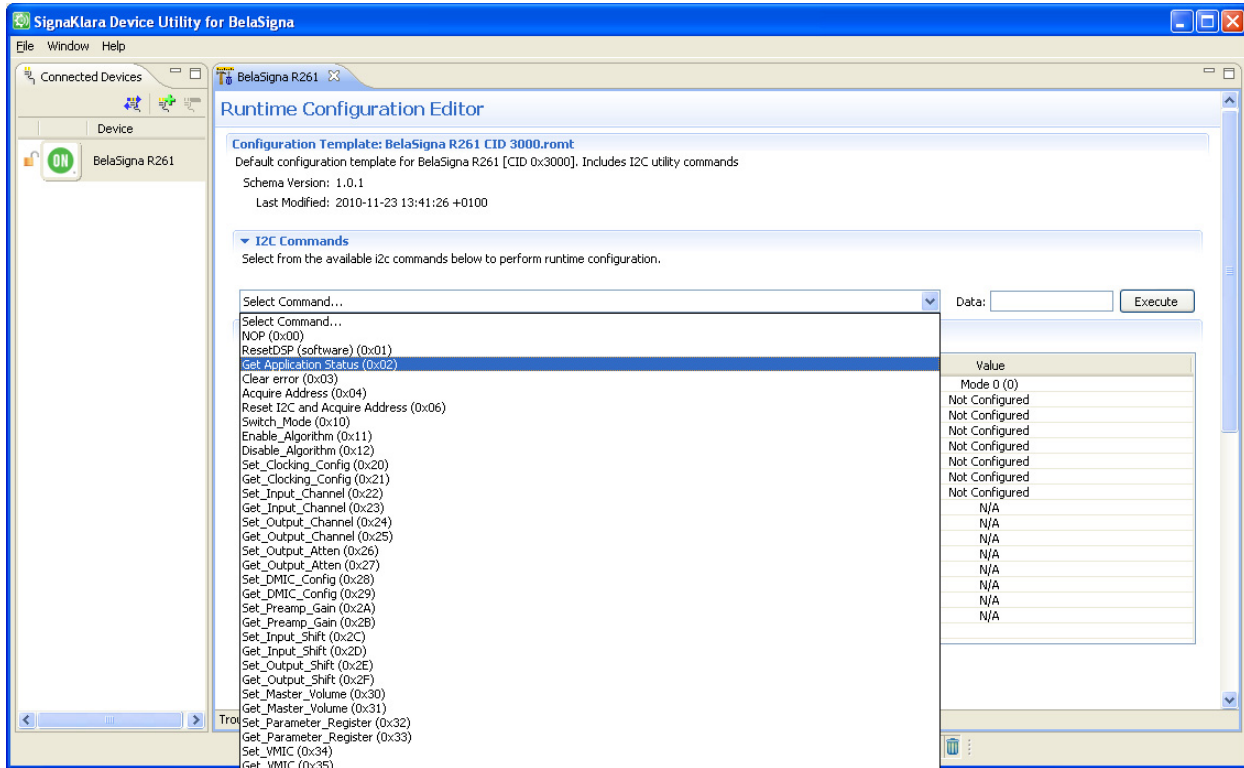


Figure 8.



## AND8485

This function does not require any argument, so you can simply click the Execute button after selecting the command in the pull-down menu. The SKDU will return the application status as follows:

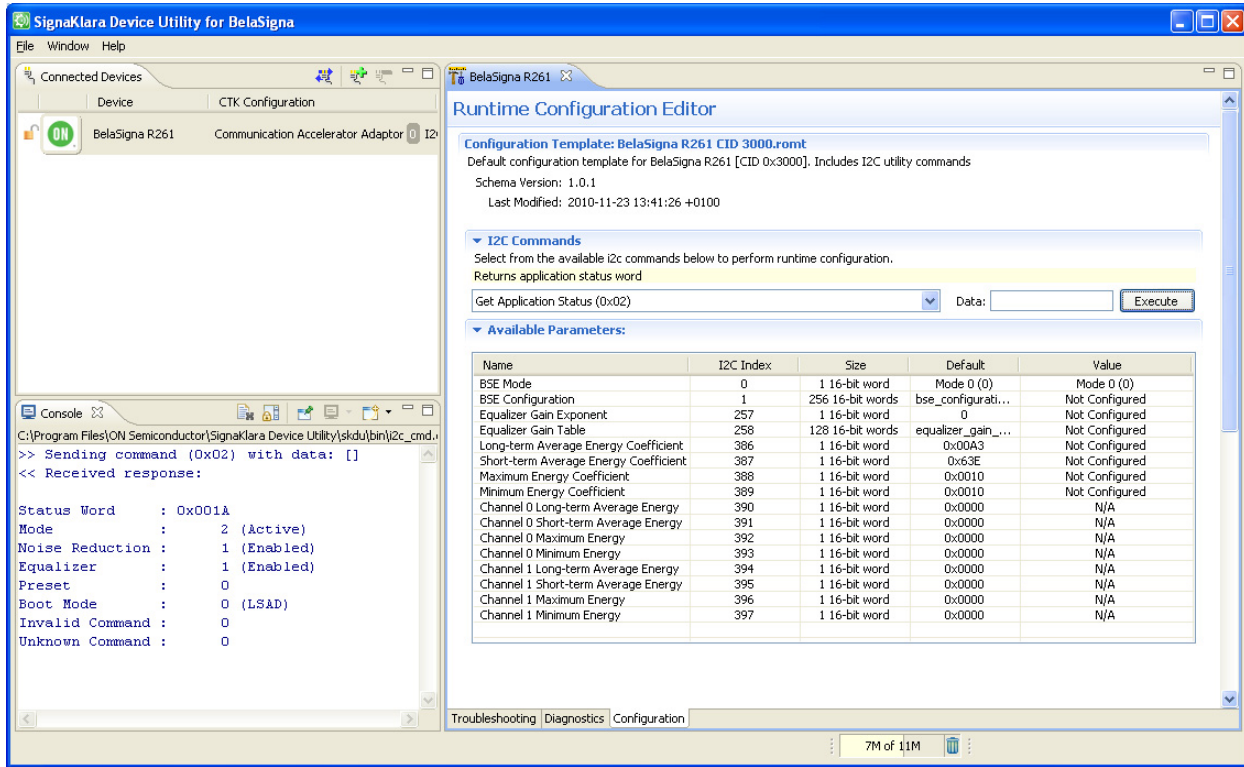


Figure 9.

The response tells you the following information:

- The current operating mode is 2 (Active)
  - Possible values could also be 0 (Stand-by), 1 (Deep Sleep), 2 (Active), 3 (Lineout) or 4 (Bypass). The default application always boot in Active Mode. See BelaSigna R261's Communications and Configuration Guide for detailed information on these various operating modes.
  - To change mode, you need to use the "Switch\_Mode" command
- The noise reduction algorithm is enabled
  - You can use the "Enable\_Algorithm" and the "Disable\_Algorithm" I2C commands to affect this parameter
- The Equalizer is enabled
  - You can also use the "Enable\_Algorithm" and the "Disable\_Algorithm" with different arguments to affect this parameter
- The current Preset is #0
  - This means that the system did boot from Preset #0, which corresponds to 2.048 MHz input clock and the BSE algorithm running in its far-talk mode (Mode0). Note that this is only true when the booting method is LSAD. When booting from EEPROM, the application that is inside EEPROM might change the application settings, but this preset will still be shown by the SKDU.
- The booting method was LSAD
  - No EEPROM content was found, so the LSAD boot method was used.
  - There have been no invalid or unknown commands sent to the device.

### Dynamically Change the Configuration of Your BelaSigna R261, Using the Runtime Configuration Editor

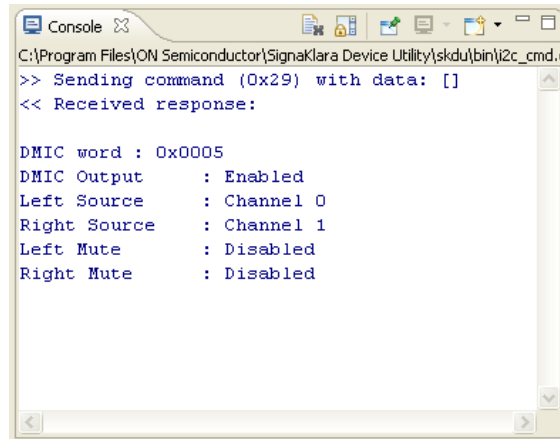
In addition to the few commands that were presented in the previous section, a number of BelaSigna R261 parameters can be modified using the runtime configuration editor in the SKDU. A few examples are discussed here. More detailed information can be found in the BelaSigna R261 Configuration and Communications Guide.

For the next steps, you have to make sure that you have a way to record the output of BelaSigna R261, no matter which output stage you are using. The DMIC interface will be used in this application note.

## AND8485

In your recording software, ensure you are recording on the proper audio device, and that you have selected a stereo recording, at a sampling rate of minimum 16 kHz.

You can observe that the left channel is processed, and that the right channel is unprocessed. We will start by checking the current DMIC configuration of BelaSigna R261, by using the “Get\_DMIC\_Config” function. Select this function in the pull-down menu and click execute. You will get the following response in the console window:

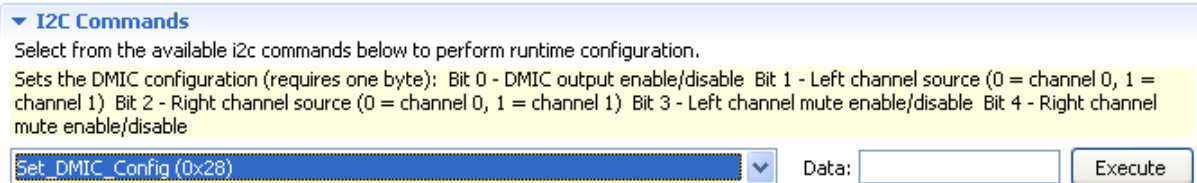


```
C:\Program Files\ON Semiconductor\SignaKlara Device Utility\skdu\bin\i2c_cmd.  
>> Sending command (0x29) with data: []  
<< Received response:  
  
DMIC word : 0x0005  
DMIC Output      : Enabled  
Left Source      : Channel 0  
Right Source     : Channel 1  
Left Mute        : Disabled  
Right Mute       : Disabled
```

**Figure 10. Check Current DMIC Configuration**

The DMIC interface is currently enabled, the left source is channel 0 (BSE processed signal) and the right source is channel 1 (unprocessed channel). You can also see that the mute functions are both disabled.

You will now use the “Set\_DMIC\_Config” function in the I2C command pull-down menu and select a particular configuration where the left source is muted, and the right source contains the processed output. To do this, check the description of this function as shown in the SKDU (see Figure 10). This function requires one byte of data, with the bits as described in the function description.

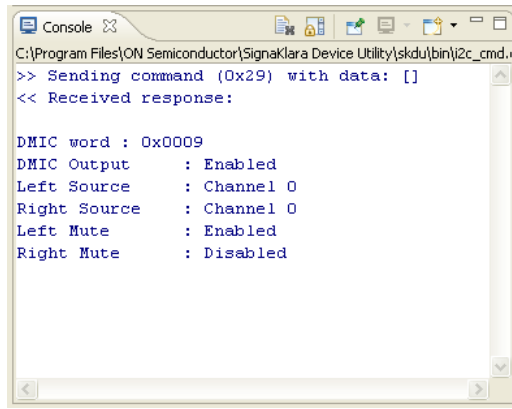


**Figure 11. Set DMIC Config Function Dialog**

You need to ensure that:

- Bit0 is set to enable the DMIC interface
- Bit1 does not matter as we will mute the left channel
- Bit2 is reset such that the right channel gets channel 0 (processed)
- Bit3 is set to enable the mute function on the left channel
- Bit4 is reset to disable the mute function on the right channel.

You can therefore type the following argument in the Data field of the “Set\_DMIC\_Config” command: 0b01001, then click Execute. Then, you may run the “Get\_DMIC\_Config” command to verify your actions, which should result in the response as shown in the following picture:



**Figure 12. Verify Your DMIC Configuration Changes**

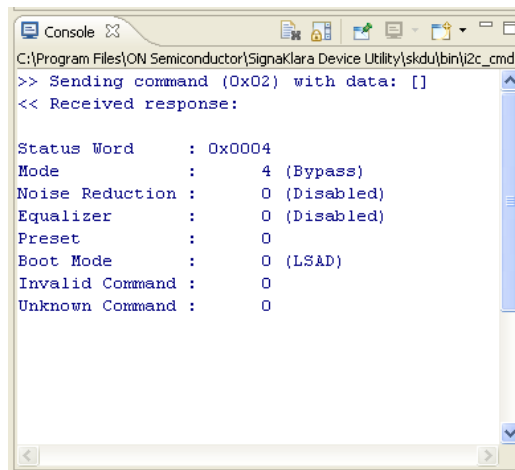
Check your recording software and verify the desired effect on the output signal.

For all the I2C commands in this pull-down menu, you can operate in a similar way, checking the description for each command, and format your arguments properly. You can use either a binary notation (e.g. 0b1001) or an hexadecimal notation (e.g. 0xFE), or a decimal representation (e.g. 7). Then, verify your changes with the appropriate read command. All configuration changes from this pull-down menu take effect immediately once you execute the I2C command.

**Check the Recording Level and Modify BelaSigna R261 Output Volume Using Bypass Mode**

You have now made a few recordings with your recording software, and you may have seen an extremely saturated signal, that may lead you to think that the algorithm performance is not good. This section will show you how to use the Bypass operating mode and its diagnostic capabilities to verify that the gain distribution between BelaSigna R261 and your recording software is properly set, to ensure there is no risk of clipping or saturation in your recordings.

First, you will need to use the “Switch\_Mode” command to have BelaSigna R261 enter its Bypass operating mode. The argument to use with this command is ‘4’, as shown in the command description. Click on the Execute button and follow-up by sending the “Get\_Application\_Status” command to verify that the device is indeed in Bypass Mode, as seen in the next figure:



**Figure 13. Confirm Bypass Mode Operation**

If you check your recording, you will now observe that your two channels are identical. Bypass Mode actually performs a full-stereo pass-through, so both microphones signal are simply transmitted to the output. Your stereo recording shows the signals from the two microphones. Note that they have been amplified by the preamplifier gains, but that there was no signal processing at all performed on them.

BelaSigna R261 offers diagnostic functions that we will now use to the audio path throughout the system. One of these functions allows the generation of a sine wave inside the output FIFOs of BelaSigna R261. The level of this sine wave is 6dB below full scale. To activate this sine wave generation, you will need to use the “Set\_Parameter\_Register” I2C command. The

## AND8485

parameter that controls this diagnostic feature is located at address 0x18E in the register file of BelaSigna R261. This parameter can take the following values:

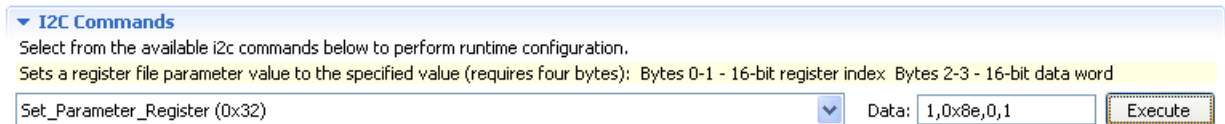
- 0: The amplified microphones are passed through the system, effectively implementing the Bypass operation
- 1: Outputs the tone signal (6dB below full scale)
- 2: Write zeroes in the output FIFO

Program the parameter register at address 0x18E of your BelaSigna R261 with the value “1”, so that the tone signal will be produced. See the following picture on how to properly format your function argument for both the parameter address and the value you want to write:



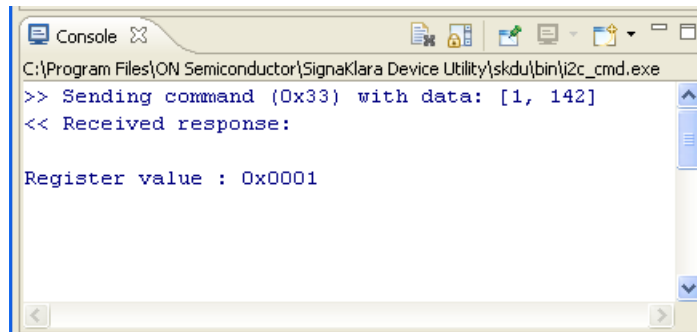
**Figure 14. Set Parameter Register, Long form**

Note that the following short form for the argument is also accepted. Only the “0x8E” needs to have the hexadecimal syntax, but the others do not, since their decimal value is identical to the hexadecimal format.



**Figure 15. Set Parameter Register, Short Form**

To verify that your action was understood by the device, issue a “Get\_Parameter\_Register” function, with the argument “1, 0x8E”. The function will return the parameter value as shown in the following picture:



**Figure 16. Get Parameter Register Response**

You can now check your recording; you should see a stereo sine wave. You can now look at the level of this signal; it should be exactly at 6dB below full scale, if you have a unity gain throughout your audio processing chain beyond the output of BelaSigna R261. If the gain is not unity, you can adjust the recording level of your computer until you can get to unity gain.

An alternative is to change the output volume of BelaSigna R261 such that any selection of the recording level on the computer would produce no saturation, but care must be taken when doing so, as you may end up attenuating the signal too much in BelaSigna R261. Re-amplifying the attenuated signal on the computer might increase the overall noise floor and decrease the performance of the BSE algorithm.

To modify the volume level of BelaSigna R261 output, you can either use the “Set\_Master\_Volume” function, which offers the full 16-bit range from a mute operation (0x0000), to unity gain (0x7FFF). Any value in between will implement an attenuation of the output signal.

Alternatively, you can use the “Set\_Output\_Shift” function which lets you define the number of 6dB shifts that BelaSigna R261 will apply on the output signal. You can obviously use a combination of both mechanisms to get to your desired output level. Note that a positive number of shifts will cause an amplification of the output signal, whereas a negative number of shifts will cause attenuation.

Note that these two parameters have to be set independently for each output channel. Also, note that these changes apply to the current operating mode, which in this case is Bypass Mode. BelaSigna R261 keeps a different set of parameters for each mode, so if you now decide to switch back to Active Mode, using the “Switch\_Mode” command, then you will need to run the same functions again to achieve the same volume modification in Active Mode. However, if you switch back again to Bypass Mode, the volume settings you applied when you were previously in Bypass Mode will still be active.

You can safely re-use the volume settings you obtained in Bypass Mode to the Active Mode, so that further recordings made in Active Mode with the BSE algorithm running will consistently provide similar output signal amplitudes in your recording software.

**Update the Algorithm Mode**

There are a number of other parameters that can also be changed, but these are not accessible from the same pull-down menu. They are in fact algorithm parameters that only get loaded when the operating mode is loaded. These parameters are available as seen on the following figure:

▼ Available Parameters:

Name	I2C Index	Size	Default	Value
BSE Mode	0	1 16-bit word	Mode 0 (0)	Mode 0 (0)
BSE Configuration	1	256 16-bit words	bse_configurati...	Not Configured
Equalizer Gain Exponent	257	1 16-bit word	0	Not Configured
Equalizer Gain Table	258	128 16-bit words	equalizer_gain_...	Not Configured
Long-term Average Energy Coefficient	386	1 16-bit word	0x00A3	Not Configured
Short-term Average Energy Coefficient	387	1 16-bit word	0x63E	Not Configured
Maximum Energy Coefficient	388	1 16-bit word	0x0010	Not Configured
Minimum Energy Coefficient	389	1 16-bit word	0x0010	Not Configured
Channel 0 Long-term Average Energy	390	1 16-bit word	0x0000	N/A
Channel 0 Short-term Average Energy	391	1 16-bit word	0x0000	N/A
Channel 0 Maximum Energy	392	1 16-bit word	0x0000	N/A
Channel 0 Minimum Energy	393	1 16-bit word	0x0000	N/A
Channel 1 Long-term Average Energy	394	1 16-bit word	0x0000	N/A
Channel 1 Short-term Average Energy	395	1 16-bit word	0x0000	N/A
Channel 1 Maximum Energy	396	1 16-bit word	0x0000	N/A
Channel 1 Minimum Energy	397	1 16-bit word	0x0000	N/A

**Figure 17. Algorithm Parameters**

The first 8 parameters can be read as well as be written to the application, whereas the last 8 parameters can only be read. The BSE related parameters only apply to the Active Mode, which is the only mode running the BSE algorithm. The Equalizer parameters apply both to the Active Mode and the Bypass Mode; whereas the next four filter coefficients only apply to Bypass Mode. See the BelaSigna R261 Configuration and Communications Guide for more information on these parameters.

Our next steps will now be to modify the BSE parameters from the runtime configuration editor of the SKDU.

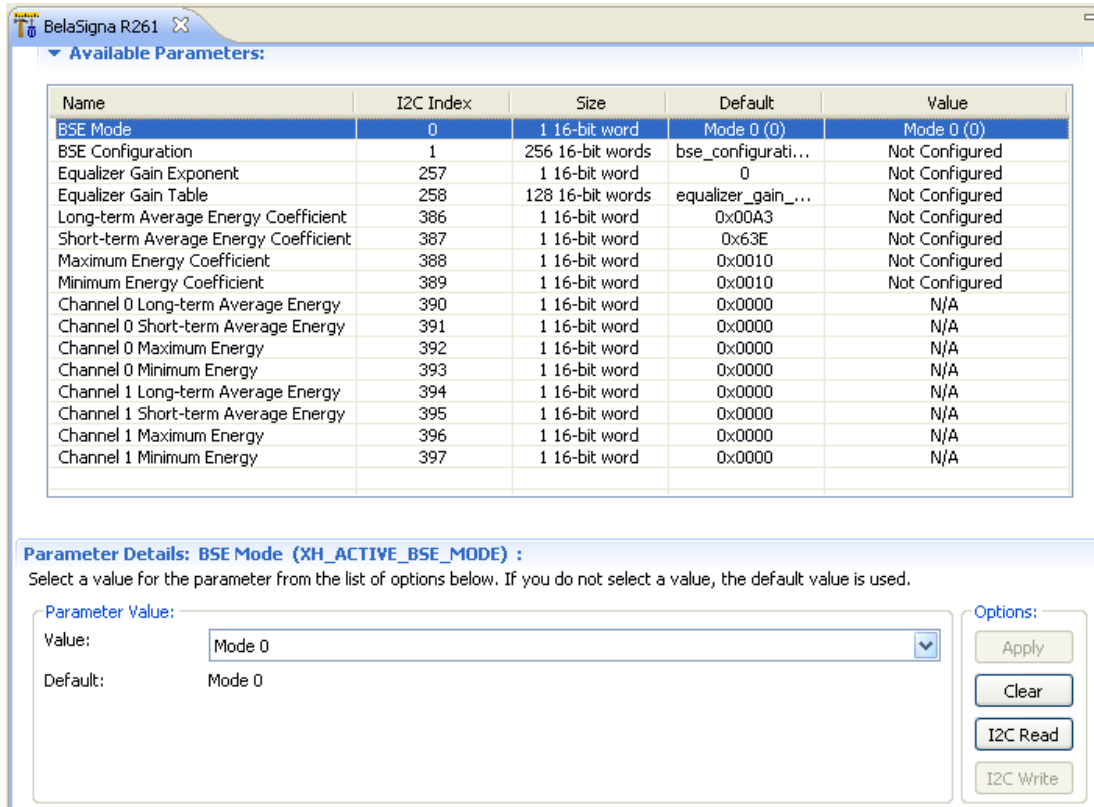
As a first example, we will now change the BSE Mode. The BSE algorithm is available as part of the ROM application in two different flavors: far-talk mode (Mode0) and close-talk mode (Mode1). A custom mode is also available, to be used when you upload a custom parameter file using the BSE Configuration parameter.

This algorithm mode configuration is loaded by the BSE algorithm when entering Active operating mode, so if you wish to change the BSE Mode, you will need to use the “Switch\_Mode” command to switch BelaSigna R261 to Active Mode with your new settings. You will need to use the “Switch\_Mode” command even if you make changes to the algorithm mode when BelaSigna BR261 is already in Active operating mode.

As seen previously in this application note, the LSAD booting method has selected preset #0, which indicates that the algorithm has start running in its far-talk mode (Mode0). We will now describe the procedures to change the algorithm mode to close-talk mode (Mode1), and then to custom mode using a custom parameter file (.d file).

## AND8485

In the BelaSigna R261 view of the SKDU, under “Available Parameters”, click on the BSE Mode parameter and you will see the dialog as shown on the following figure:



**Figure 18. BSE Mode Dialog**

Begin by issuing an I2C Read command. It will update the value of this parameter with the BSE Mode currently being used by BelaSigna R261. Then, select Mode 1 from the drop-down list, and click Apply. The I2C Write button will become active and you can click on it to send this new parameter to BelaSigna R261. Issue another I2C Read command to verify that the operation succeeded.

As mentioned earlier, even though this parameter was updated on the device, the application is still running with the previous setting and you will need to issue a “Switch\_Mode” command to let the BSE algorithm re-start its operation in the new mode. Select the “Switch\_Mode” command from the list of I2C commands, use ‘2’ as the argument (Active Operating Mode), and click the Execute button. You can now verify using your recording software that the BSE algorithm mode has now changed to Mode1 (close-talk).

The third available algorithm mode in BelaSigna R261 is called “Custom Mode”, which must be used in conjunction with a custom parameter file (.d file). The .d file contains 256 parameters, that are AES encrypted by ExAudio (the developer of the BSE algorithm), and that will allow the deployment of custom algorithm tuning. One potential custom mode that you may wish to use is an algorithm parameter set developed for BelaSigna R261 that is called Near-Talk mode. Near-Talk mode can be considered as an intermediate BSE configuration between far-talk and close-talk. We will now see how you can put your device in Near-Talk mode, using the custom parameter file.

## AND8485

We will start by uploading the new set of 256 BSE algorithm parameters. Select the BSE Configuration item under “Available Parameters”, and see the new dialog that appears, as in the following figure.

The screenshot shows a software interface for configuring BSE parameters. At the top, a section titled "Available Parameters:" contains a table with the following data:

Name	I2C Index	Size	Default	Value
BSE Mode	0	1 16-bit word	Mode 0 (0)	Mode 1 (1)
BSE Configuration	1	256 16-bit words	bse_configurati...	Not Configured
Equalizer Gain Exponent	257	1 16-bit word	0	Not Configured
Equalizer Gain Table	258	128 16-bit words	equalizer_gain_...	Not Configured
Long-term Average Energy Coefficient	386	1 16-bit word	0x00A3	Not Configured
Short-term Average Energy Coefficient	387	1 16-bit word	0x63E	Not Configured
Maximum Energy Coefficient	388	1 16-bit word	0x0010	Not Configured
Minimum Energy Coefficient	389	1 16-bit word	0x0010	Not Configured
Channel 0 Long-term Average Energy	390	1 16-bit word	0x0000	N/A
Channel 0 Short-term Average Energy	391	1 16-bit word	0x0000	N/A
Channel 0 Maximum Energy	392	1 16-bit word	0x0000	N/A
Channel 0 Minimum Energy	393	1 16-bit word	0x0000	N/A
Channel 1 Long-term Average Energy	394	1 16-bit word	0x0000	N/A
Channel 1 Short-term Average Energy	395	1 16-bit word	0x0000	N/A
Channel 1 Maximum Energy	396	1 16-bit word	0x0000	N/A
Channel 1 Minimum Energy	397	1 16-bit word	0x0000	N/A

Below the table is a section titled "Parameter Details: BSE Configuration (XH\_AES\_INPUTW) :". It contains the following text: "This parameter is defined using external data. Specify the location of the relative data file (\*.d) that will provide the data for this parameter. The file must contain the correct number of words, of the correct size, corresponding to the size of the parameter."

The "Parameter Value:" section includes a "File:" input field with a "Select..." button and a "Default:" field containing the text "bse\_configuration.d".

The "Options:" section contains four buttons: "Apply", "Clear", "I2C Read", and "I2C Write".

**Figure 19. BSE Configuration Dialog**

Locate the “neartalk.d” file in the folder that is distributed with this application note. Practically, you can place this file anywhere on your hard drive.

In the Parameter Details view, click on the Select button, and locate the file. Click Apply, and then click I2C Write. This will download the 256 parameters in the configuration file to the device.

Before issuing another “Switch\_Mode” command, you will need to instruct the BSE algorithm to use this custom parameter file. The way to do this is by changing the BSE Mode parameter to “Custom Mode”. Follow the same steps as described above to configure BelaSigna R261 in custom mode.

Then, issue the mode switch command to have the BelaSigna R261 restart in Active Mode. Then you will be able to check your recordings and experience this Near-Talk mode.

### Define Custom Equalizer Gains

If you want to modify the frequency shape of BelaSigna R261 output, you can use the built-in equalizer with a custom gain file. By default, BelaSigna R261 is executing the equalizer algorithm with a unity gain table. Note that the equalizer is considered as an algorithm by BelaSigna R261, as explained earlier; consequently, it is controlled by an Enable/Disable flag in the application. The “Enable\_Algorithm” and “Disable\_Algorithm” can be used to control the equalizer. The equalizer status can always be monitored by the “Get\_Application\_Status” I2C command.

We will now assume that your equalizer is enabled, and that you have edited the “equalizer\_gain\_table.d” file, that was provided as part of the software package with this application note, to implement your desired frequency shaping.

To provide this new equalizer gain table to BelaSigna R261, follow the same steps as for defining new custom BSE parameter file, as explained above. Select the “Equalizer Gain Table” parameter, in the parameter details, browse your computer to select the new gain file and select it. Click Apply, and I2C Write. Remember that you also need to issue a new “Switch\_Mode” I2C command to restart Active Mode, to have the new equalizer take effect.

The equalizer algorithm not only uses the gain table discussed above, but it also uses a “Gain Exponent”. This parameter can also be updated from the parameter view; it is a 16-bit value that defines the number of 6 dB shift to apply to the gain table. Care must be taken when adding gain, such that it doesn’t saturate BelaSigna R261’s output signal.

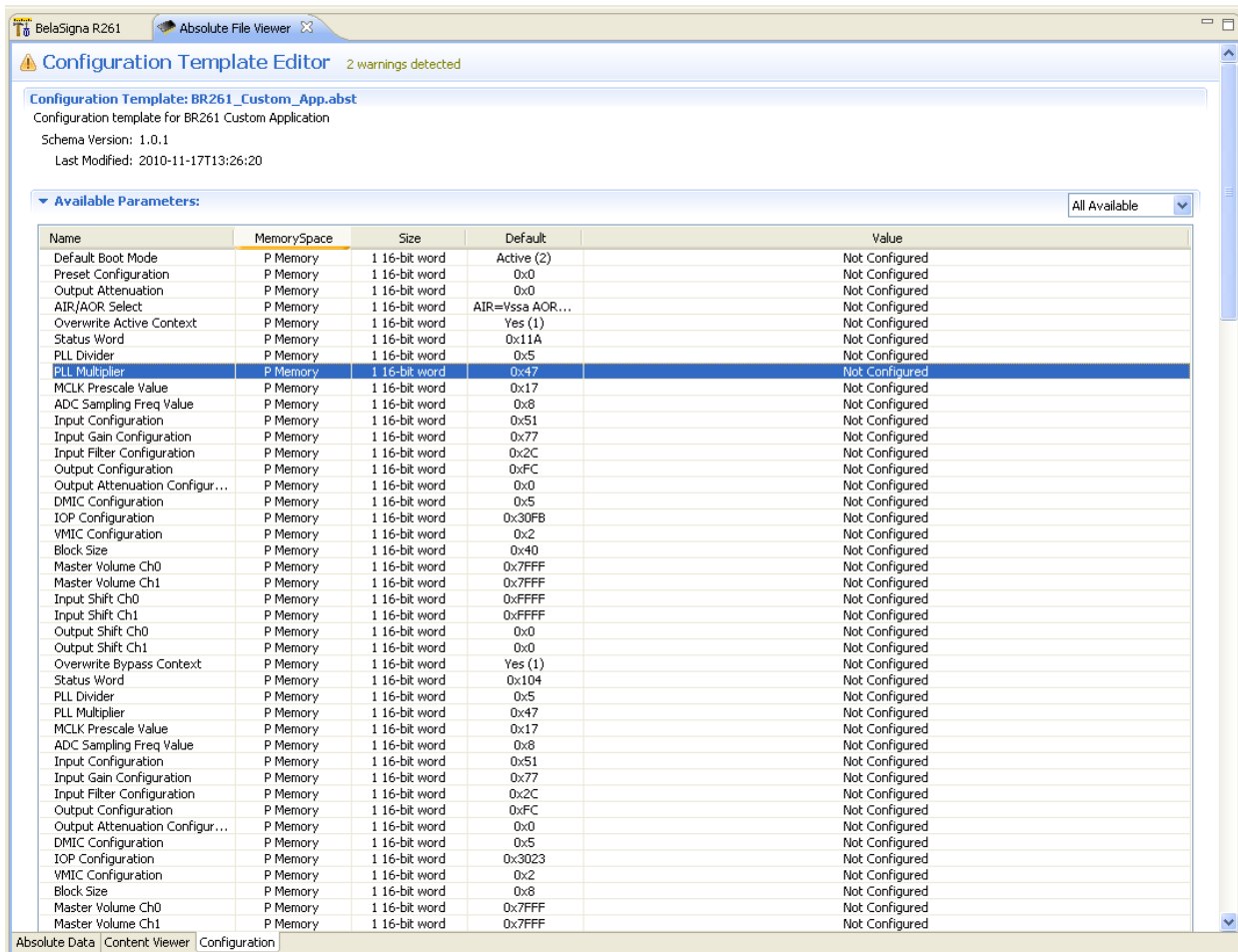
**Create a Custom Application With Your Desired Settings**

Now that you have experienced the performances of the ROM algorithm, and used the runtime configuration editor of the SKDU to configure your desired application settings, we will show you how you can generate both a .abs file as well as a .eeprom file using the SKDU, so that you can avoid the manual configuration procedure.

For this purpose, we are introducing the concept of an “application template”. This consists of an .abst file that can be obtained from ON Semiconductor. The .abst file stores information about the configuration settings of a corresponding .abs file, allowing you to modify these settings in the SKDU.

The process consists of loading the .abst file in the SKDU and performing all your custom configuration steps “off–line” within the template editor. “Off–line” means that an active I2C connection to BelaSigna R261 is not required. You will see how you can generate a new .abs file as well as a new .eeprom file that will have all your custom settings.

In your SKDU, select File→ Open, and browse your computer for the BR261\_Custom\_App.abst file that you obtained as part of the software package distributed with this application note. You will see a new view open beside the BelaSigna R261 view that you have not seen so far. This new view looks like the one shown in the following picture.



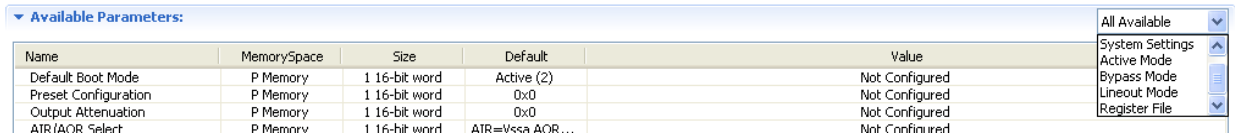
**Figure 20. Custom Application Template Editor**

In the BelaSigna R261 view, you have been able to dynamically interact with your BelaSigna R261 via the I2C interface, whereas with this new view, there is no capability to interact with the device. It is there to allow the customization and the creation of a new custom application.



## AND8485

All the parameters that are available for customization in a custom application are shown in this dialog. Furthermore, they can be filtered in different groups as shown in the following picture:



Name	MemorySpace	Size	Default	Value
Default Boot Mode	P Memory	1 16-bit word	Active (2)	Not Configured
Preset Configuration	P Memory	1 16-bit word	0x0	Not Configured
Output Attenuation	P Memory	1 16-bit word	0x0	Not Configured
ATR/AOR Select	P Memory	1 16-bit word	ATR=Vcca AOR...	Not Configured

Figure 21.

When selecting “All Available” from the drop-down list, you will see all the parameters. The other filters for different types of parameters are defined as follows:

- System Settings
  - These parameters are valid all the time, and will always apply regardless of the operating mode.
  - You can typically define in which operating mode the application will start. If you want to have an application that immediately starts in Bypass Mode, you can configure it here.
  - You can instruct the application to load the settings from a selected preset, which would essentially be identical to configuring the voltage divider on the CONFIG\_SEL pin, and booting from the LSAD configuration instead of EEPROM. This parameter allows selecting a predefined mode.
  - You can define a global analog output attenuation that will apply in all operating modes.
  - Lastly, you can select the analog input and output reference signal. This parameter usually does not need to be modified, and can be left as default.
- Active Mode / Bypass Mode / Lineout Mode settings
  - Each operating mode is defined by a “Context Table”. The parameters shown under each mode are the same, but can have different values for each operating mode (active, bypass and lineout). For example, and to illustrate the master volume control discussion earlier in this application note, you can have a different master volume configuration in Bypass Mode compared to Active Mode.
  - The ROM application already has a context table for each mode, but a custom application allows you to overwrite the built-in settings. Each mode has a parameter called “Overwrite <Mode> Context”. When this parameter is set, the custom application will always overwrite the ROM context with the custom context table. If the parameter is reset, then, the ROM context table will be used.
  - It is important to note that when the “Overwrite” flag is set, the “Preset Configuration” as defined in the system settings will not be used. Therefore, the “Preset Configuration” only applies to modes which do not have their “Overwrite” flag set. By default, the “Overwrite” flags for all modes are set, so you will effectively need to reset them, if you want the “Preset Configuration” to be taken into account.
  - The operating mode context tables allow you to set various parameters. All the parameters in the context tables can be selected, and the corresponding dialog will appear, as can be seen in the following picture.

Available Parameters: Active Mode

Name	MemorySpace	Size	Default	Value
Overwrite Active Context	P Memory	1 16-bit word	Yes (1)	Not Configured
Status Word	P Memory	1 16-bit word	0x11A	Not Configured
PLL Divider	P Memory	1 16-bit word	0x5	Not Configured
PLL Multiplier	P Memory	1 16-bit word	0x47	Not Configured
MCLK Prescale Value	P Memory	1 16-bit word	0x17	Not Configured
ADC Sampling Freq Value	P Memory	1 16-bit word	0x8	Not Configured
Input Configuration	P Memory	1 16-bit word	0x51	Not Configured
Input Gain Configuration	P Memory	1 16-bit word	0x77	Not Configured
Input Filter Configuration	P Memory	1 16-bit word	0x2C	Not Configured
Output Configuration	P Memory	1 16-bit word	0xFC	Not Configured
Output Attenuation Configur...	P Memory	1 16-bit word	0x0	Not Configured
DMIC Configuration	P Memory	1 16-bit word	0x5	Not Configured
IOP Configuration	P Memory	1 16-bit word	0x30FB	Not Configured
VMIC Configuration	P Memory	1 16-bit word	0x2	Not Configured
Block Size	P Memory	1 16-bit word	0x40	Not Configured
Master Volume Ch0	P Memory	1 16-bit word	0x7FFF	Not Configured
Master Volume Ch1	P Memory	1 16-bit word	0x7FFF	Not Configured
Input Shift Ch0	P Memory	1 16-bit word	0xFFFF	Not Configured
Input Shift Ch1	P Memory	1 16-bit word	0xFFFF	Not Configured
Output Shift Ch0	P Memory	1 16-bit word	0x0	Not Configured
Output Shift Ch1	P Memory	1 16-bit word	0x0	Not Configured

Parameter Details: Master Volume Ch0 (P\_ACTIVE\_CONTEXT\_MASTER\_VOLUME\_CH0) : Active Mode

Specify a numeric value for the parameter that falls between the specified minimum and maximum values. If you do not specify a value, the default value is used. Note: separate multiple word values with a comma.

Parameter Value:

Options:

Value:   
 Default: 0x7FFF   
 Minimum: 0   
 Maximum: 32767

Figure 22. Active Mode Context Table

- You have probably noted that all these parameters must be given a specific value, and that no further information is provided on what kind of values to specify depending on the configuration you want to apply. For additional information, please refer to the BelaSigna R261 Configuration and Communications Guide.
- Register File settings
  - These parameters concern the algorithms running in the different operating modes. As discussed previously, some of these parameters apply to Active Mode, some apply to Bypass Mode.

As an example, let us implement the following changes to the template, before we export this new configuration into a new EEPROM image:

- In System Settings, ensure the default boot mode is Active:
  - Select System Settings in the pull-down menu
  - Click on Default Boot Mode
- In the parameter details, select “Active” from the pull-down menu, and click apply. You can now see that the template was modified to include your selection:

Available Parameters: System Settings

Name	MemorySpace	Size	Default	Value
Default Boot Mode	P Memory	1 16-bit word	Active (2)	Active (2)
Preset Configur...	P Memory	1 16-bit word	0x0	Not Configured

Figure 23. Default Boot Mode Configured

- In Active Mode Settings:
  - Reverse the two output channels: enter 0x3 for the DMIC Configuration
  - Define the master volume to the value of 0x5000 for both channels
- In Bypass Mode Settings:
  - Reverse the two output channels: enter 0x3 for the DMIC Configuration
  - Define the master volume to the value of 0x5000 for both channels
- In the Register File Settings:
  - Set the BSE Mode to “Custom Mode”
  - Set the “Overwrite BSE Parameters” to “Yes”
  - Select the Near Talk BSE configuration file (bse\_neartalk.d)
  - Set the Bypass Diagnostic Mode to “Output Tone”.
    - With this setting, any time you enter Bypass Mode, BelaSigna R261 will immediately output the tone, scaled by the Bypass volume control.

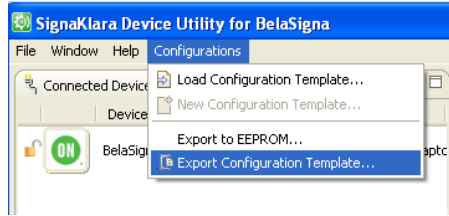
You now have completed the configuration of the template! Use “File → Save As” to save a copy of your configuration template. Be sure to avoid overwriting your original template, so that you will be able to reuse it for future customizations you will want to do by restarting from the original template.

**Create an EEPROM Application With Your New Application**

After successfully creating a custom template, the next steps will be to use this new template to create the necessary files for downloading to your BelaSigna R261. We assume that you have an external EEPROM attached to BelaSigna R261, and that BelaSigna R261 is already configured properly to boot from EEPROM. At this point, with the latest version 1.2 of the SKDU, you can either decide to create a new .abs file, but you can also decide to immediately create an EEPROM image. For information purposes, we will start by showing you how you can create a custom .abs file

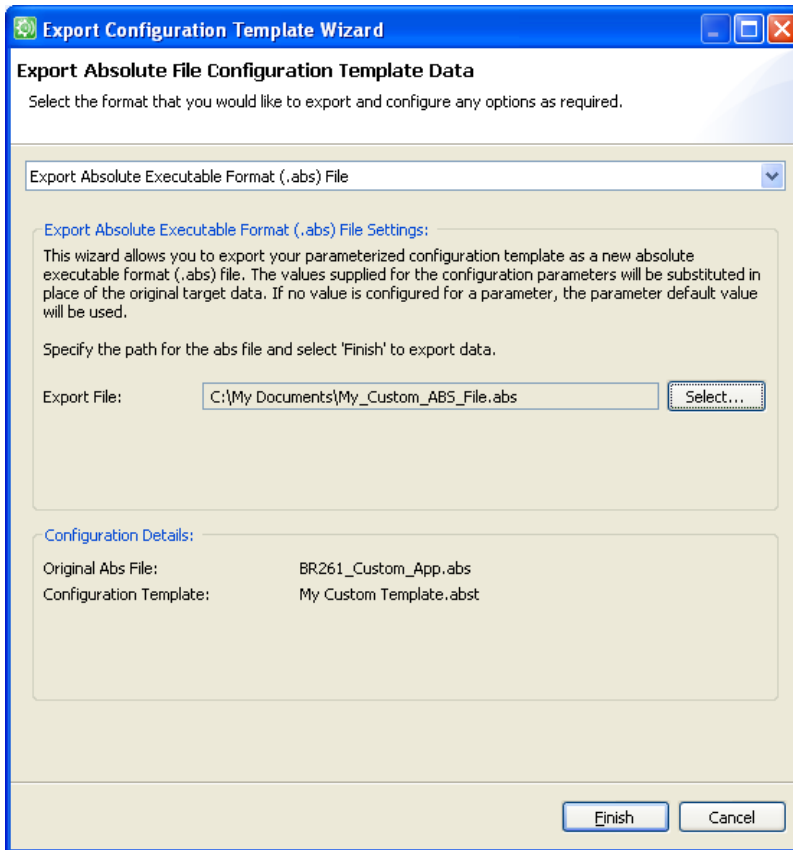
First, export the template.

- From the Configurations menu, select Export Configuration Template...



**Figure 24. Exporting an ABS File**

- Fill-in the wizard by selecting a new custom ABS file name. This will be your new .abs file. You can see in the following picture that the SKDU will use the original executable file “BR261\_Custom.abs” and your new template “My Custom Template.abst” to generate the new “My Custom ABS File.abs”



**Figure 25. Export Configuration Template Wizard**

- Click Finish.
- You may see a warning telling you that you have not provided any valid value for the Equalizer Gain Table; you can safely ignore this warning and click ok.
- Now you have a new .abs File located in the folder you have specified.

## AND8485

We will now see how to download this .abs file directly to BelaSigna R261:

- Switch to the Advanced View of the SKDU by using the Menu item Window → Show Advanced
- The “Application Shortcuts” section in the SKDU Explorer view that just appeared is showing a link to your newly created .abs file. Your SKDU window should now look like this:

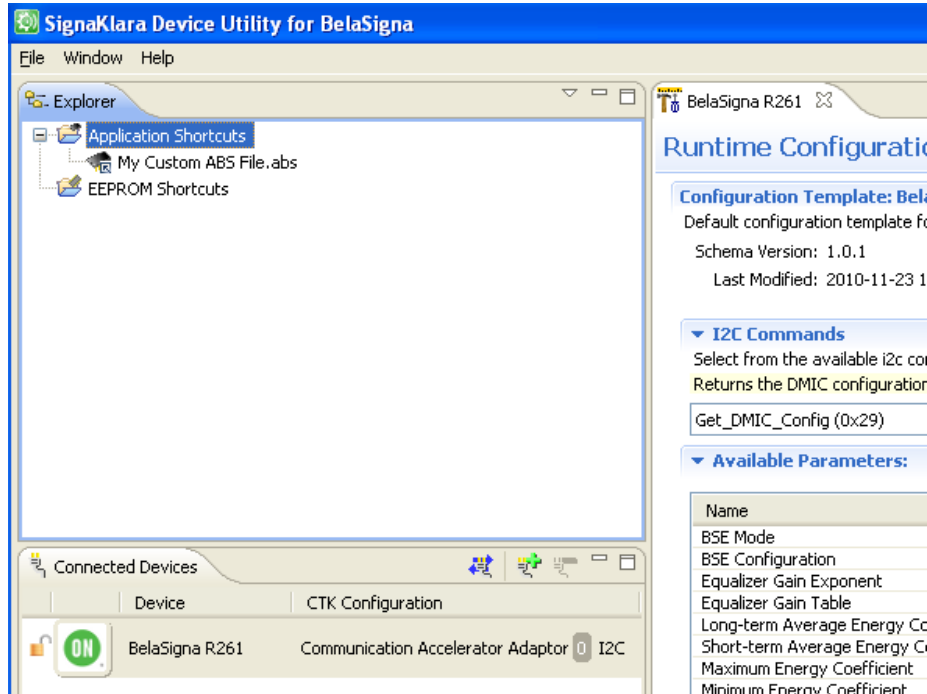


Figure 26. Creating an application shortcut

- Double click on the .abs file
- Click “Download File to Device” and select your BelaSigna R261 device
- Click Finish
- Confirm the download operation by clicking OK.

Your new application has now been downloaded on your BelaSigna R261. To see it running, follow these steps:

- Switch back to the “BelaSigna R261” window in your SKDU to allow real-time communication with the application.
- Issue a “Reset DSP Software” I2C command from the BelaSigna R261 view in the SKDU. Note that you should not issue a “Go 0x0000” command, as this would restart a complete booting operation, and would not run your application at all.
- Select the “BSE\_Mode” parameter under “Available Parameters”, and click on the “I2C Read” button. You can then verify that your application has indeed started in custom mode.
- Run the “Get\_Master\_Volume” command, and you should see that your master volumes are now both set at 0x5000.

Now we will create an EEPROM image with this new .abs file:

- Switch back to your “Absolute File Viewer” window in the SKDU
- From the Configurations menu, select “Export to EEPROM...” as shown on the following picture:

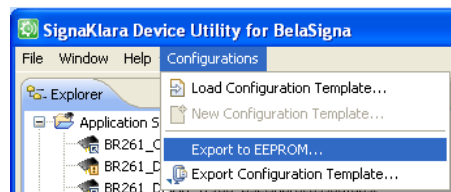


Figure 27. Export to EEPROM

- Select BelaSigna R261 in the wizard, give your new EEPROM image a name and click “Finish”
- Open your “EEPROM Shortcuts” in the Explorer view of the SKDU to see an alias to newly created .eprom file.

## AND8485

- Double click on this alias, “Download File to Device” as shown on Figure 25 and click Finish.

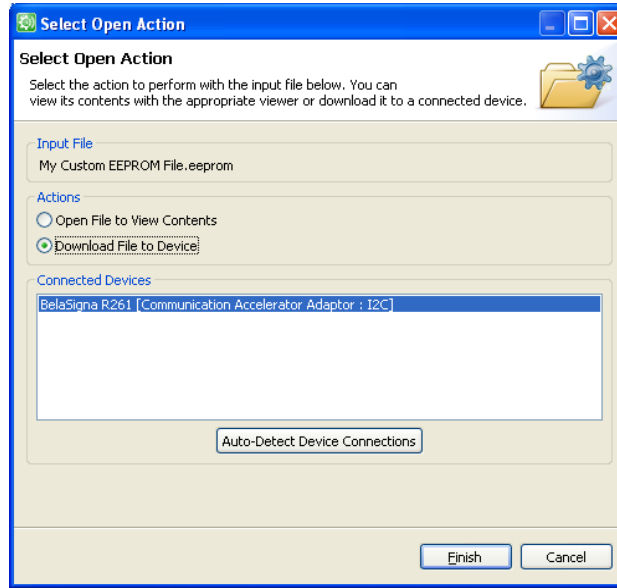


Figure 28. Download to EEPROM

- You will see a warning informing you that the content of your EEPROM will be modified, click OK to download the new application.
- You will then see the evolution of the download progress as shows on the following picture:

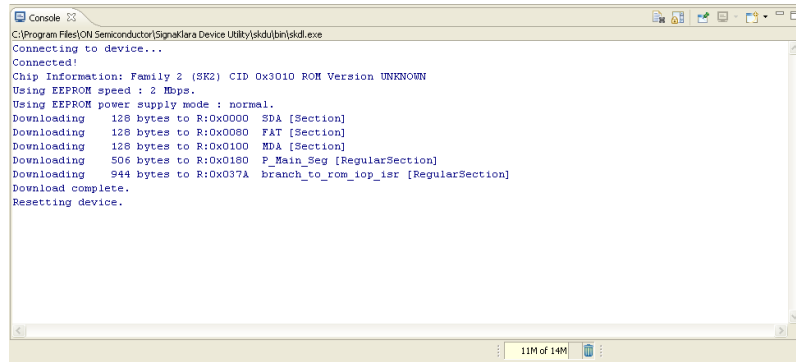


Figure 29. Download Complete

- BelaSigna R261 has been rebooted by the download process, and if your hardware was properly configured for EEPROM booting, it is now running your custom application.

### Verify That Your New Application is Running

Follow these steps below to verify that your custom application is now running it:

- You can verify this by checking your recording software, or alternatively, by issue a “Get Application Status” I2C command from the BelaSigna R261 view of your SKDU, to confirm that the booting method is now EEPROM as shown on the following picture:

```

C:\Program Files\ON Semiconductor\Signalara Device Utility\skdu\bin\i2c_cmd.exe
>> Sending command (0x02) with data: []
<< C:\Program Files\ON Semiconductor\Signalara Device Utility\skdu\bin\i2c_cmd.exe

Status Word      : 0x011A
Mode             :      2 (Active)
Noise Reduction  :      1 (Enabled)
Equalizer        :      1 (Enabled)
Preset           :      0
Boot Mode        :      1 (EEPROM)
Invalid Command  :      0
Unknown Command  :      0

```

**Figure 30. Confirm EEPROM Booting Method**

- Select the BSE\_Mode parameter, and issue an “I2C Read” command to verify that BelaSigna R261 is indeed running in custom algorithm mode
- Issue a Mode Switch command to enter Bypass Mode
- Get the application status again to verify that BelaSigna R261 is now running Bypass Mode
- Issue a Get\_Parameter\_Register with argument 0x1,0x8E to verify that the value was set to 1 (Output tone is enabled)
- Verify that your recording is indeed showing a sine wave on both channels.
- Go back to Active Mode
- Get the application status to verify that you are indeed in Active Mode
- Read the master volume to verify that it is set to 0x5000.

If you were able to verify all of the above, then you have successfully created and downloaded your own EEPROM image for BelaSigna R261.

### Using a Command Line Script to Download the EEPROM Image

You have now seen how to use the SKDU to download an EEPROM image to your BelaSigna R261, but you may need command line utilities to equip computers in the production facilities, rather than having to distribute the SKDU, and to have a more automated process. This section will provide you the necessary information to download your EEPROM image using a unique command.

In the folder that was distributed with this application note, you will see a sub-folder called “Download Scripts”. Open a DOS command prompt and navigate to the “Download Scripts” directory. Edit the file called “BR261\_EEPROM\_Write.bat” to ensure it is pointing on the .eeprom file that you want to download. Run the batch file with your device connected over the I2C port, it will download the content of the .eeprom file to your device.

### Creating an Intel HEX Format for Standard EEPROM Programmers

Your manufacturing process may force you to pre-program your EEPROMs, before they are actually mounted on the target hardware. In this case, follow the steps in this section to convert your EEPROM into this format.

In the folder that was distributed with this application note, you will see a folder called “EEPROM Conversion”; enter this folder from within a command prompt. Edit the file “Convert\_to\_HEX.bat” with a text editor, and ensure it points to the EEPROM file that you need to convert. Execute the batch file.

### How to Restore an Empty EEPROM?

This section will guide you through the steps to follow, if for any reason you would like to prevent BelaSigna R261 to boot from the SPI EEPROM, and boot by reading the LSAD on the CONFIG\_SEL pin (assuming that you cannot control this with the CONFIG\_SEL pin).

When BelaSigna R261 is looking for an EEPROM as part of its booting process, it will start by checking if an SPI EEPROM is connected or not. Depending on the hardware that you are using, you may have the possibility to disconnect the SPI EEPROM from BelaSigna R261, so in this case, you do not need to modify the content of your EEPROM, as the device will simply not see the EEPROM, and will fall back on the LSAD booting method.

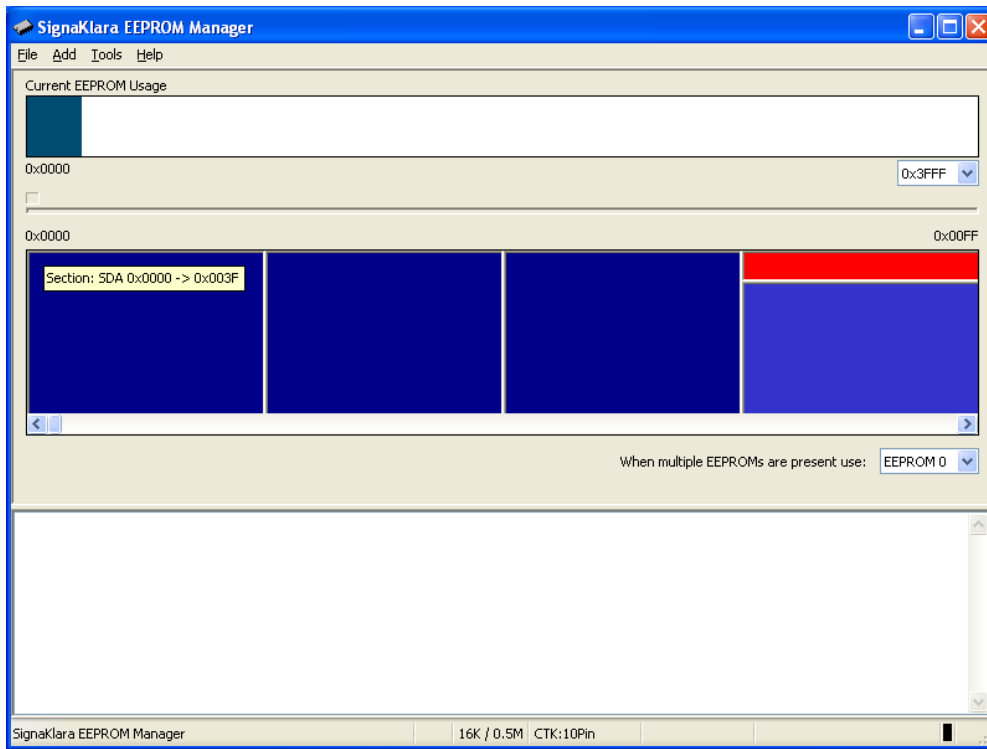
On the other hand, if you do not have the possibility to disconnect your EEPROM from BelaSigna R261, one way to avoid booting from it, is to ensure its SDA section is empty. In fact, after detecting an SPI EEPROM, BelaSigna R261 will continue the booting process by checking the SDA section inside that EEPROM. The SDA section is supposed to be formatted specifically such that the booting process can continue. Replacing an existing and valid SDA section by another one that

## AND8485

contains only zeroes will ensure that BelaSigna R261 will abort the EEPROM booting process, and will fall back on reading the LSAD on the CONFIG\_SEL pin.

Follow these steps to create another EEPROM image that has all zeroes inside the SDA section:

- In the Explorer view of your SKDU, double click on your EEPROM image
- Select “Open File to View Contents” and click Finish
- Point your cursor on the SDA section, as shown on the following picture:



**Figure 31. SDA Section in EEMGR**

- Right-click when pointing on the SDA section and select “Delete”
- Select “Add → Add Files” from the EEMGR menu
- Select the file called “SDA\_zeroes.o” provided by ON Semiconductor as part of the zip archive file distributed with this application note
- Select the option #2 (a simple section not included in the FAT) and click OK
- Ensure the proposed address is 0x0000 and click OK
- If you now double click on your new SDA section, you will see all zeroes displayed in the bottom window of the EEMGR.
- Select “File → Save As” from the EEMGR menu
- Select EEPROM binary files in the available file types
- Give your new EEPROM image a name (e.g. BR261\_Zeroes.eeprom) and click Save.
- Close your EEMGR
- From your Windows Explorer, locate your new .eeprom file, select it, drag it into the SKDU window, and drop it on the EEPROM Shortcuts item in the Explorer view
- Double click on the new shortcut, select “Download File to Device”, ensure your BelaSigna R261 device is highlighted and click Finish.
- BelaSigna R261 will automatically reboot and will then fail searching for valid EEPROM content, hence it will boot using the LSAD method.
- Issue a Get Application Status command to verify that the system has now booted in LSAD mode.

BelaSigna is a registered trademark of Semiconductor Components Industries, LLC (SCILLC).

**ON Semiconductor** and **ON** are registered trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

## PUBLICATION ORDERING INFORMATION

### LITERATURE FULFILLMENT:

Literature Distribution Center for ON Semiconductor  
P.O. Box 5163, Denver, Colorado 80217 USA  
**Phone:** 303-675-2175 or 800-344-3860 Toll Free USA/Canada  
**Fax:** 303-675-2176 or 800-344-3867 Toll Free USA/Canada  
**Email:** [orderlit@onsemi.com](mailto:orderlit@onsemi.com)

**N. American Technical Support:** 800-282-9855 Toll Free  
USA/Canada  
**Europe, Middle East and Africa Technical Support:**  
Phone: 421 33 790 2910  
**Japan Customer Focus Center**  
Phone: 81-3-5773-3850

**ON Semiconductor Website:** [www.onsemi.com](http://www.onsemi.com)  
**Order Literature:** <http://www.onsemi.com/orderlit>  
For additional information, please contact your local  
Sales Representative