AN0650

CS2600 OTP Programming Guide

Introduction

The CS2600 incorporates a customer-programmable OTP memory which can be used to automatically configure the device after power-up. The OTP memory enables the device to be factory programmed for a specific target application, removing the need for a host system to configure the device.

The OTP memory is programmed using the I2C or SPI interface. The CS2600 supports two different OTP programming methods, for production and prototyping respectively. A device may be programmed using either method, but not both.

- Prototype programming provides greater flexibility to reprogram the device during product development.
- Production programming ensures data integrity using an error correction code (ECC) algorithm.

This document describes how to program the CS2600 OTP memory using the Hazelburn development platform and SoundClear Studio (SCS) tools. It also describes how the device can be programmed in a production environment.

Table of Contents

1 Hardware Connections and OTP Programming Supply

The CS2600 requires an OTP programming supply (VDD_OTP) when programming the OTP memory. The programming supply may be provided internally or externally, depending on the system supply voltage (VDD).

- If the VDD supply is 3.3 V, the programming supply can be provided using an internal LDO regulator; in this case, the output of the regulator, VLDO_OTP, should be connected to VDD_OTP.
- If the VDD supply is 1.8 V, the programming supply must be provided externally.

The typical connections in each case are shown in Figure 1. Note that VDD must be present before enabling the VDD_OTP pin. The external VDD_OTP supply must be removed before powering down VDD.

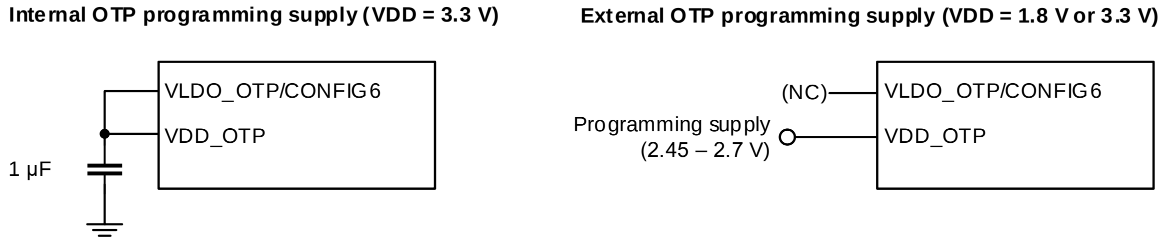

Internal OTP programming supply (VDD = 3.3 V)
VLDO_OTP/CONFIG6
VDD_OTP
1 µF
External OTP programming supply (VDD = 1.8 V or 3.3 V)
(NC)
VLDO_OTP/CONFIG6
Programming supply
(2.45 – 2.7 V)
VDD_OTP

Figure 1 OTP Programming Supply Connections

Programming the OTP memory is supported using the Hazelburn development platform (CDB-CLOCKING-MB with daughter cards). The CS2600 device is accommodated in a socket on the CDB2600-DC-SKT daughter card. The SoundClear Studio (SCS) tool is used to support OTP programming.

The Hazelburn system supports clock inputs and outputs and allows configuration and programming for a variety of possible use cases. A user guide (CDB-CLOCKING_DS1386DB1.pdf) is available for general information on the Hazelburn system.

1.1 Hazelburn Development Platform

The Hazelburn development platform is illustrated in Figure 2. The board is powered and controlled via a single USB connection. An FTDI device supports the I2C/SPI communications to control device and board via the USB connection.

Caution:

When connecting external power supplies, ensure that the supplies are disabled before connecting to the Hazelburn system.

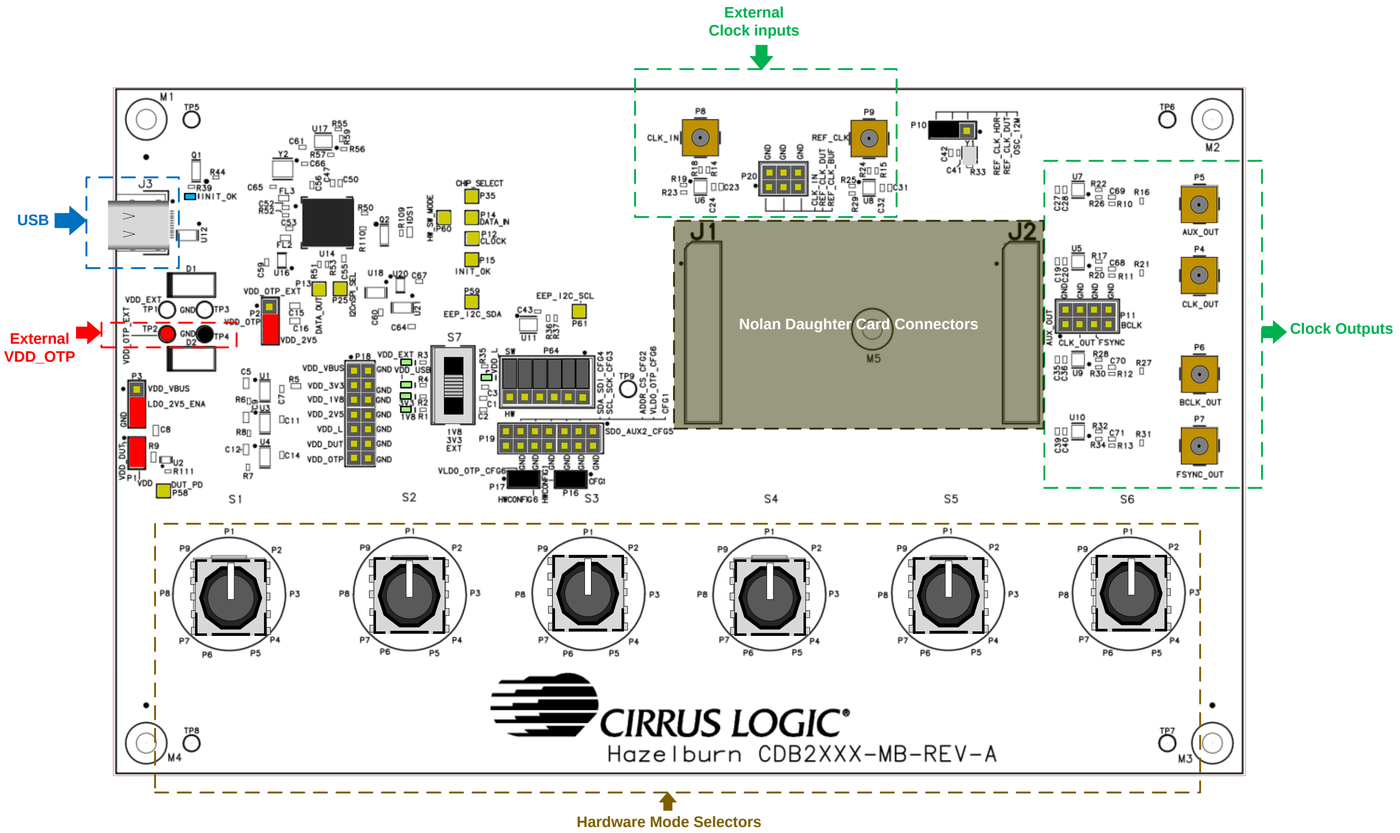


Figure 2 Hazelburn Board Overview

The Hazelburn system generates all the required power supplies from the 5 V USB supply.

The VDD supply to the CS2600 (on the daughter card) can be set to 3.3 V or 1.8 V using Switch S7. An external VDD can be provided via test point TP1 if required.

The VDD_OTP supply to the CS2600 can be provided from the USB supply, using a 2.5 V regulator. An external VDD_OTP can be provided via test point TP2 if required. The VDD_OTP is configured on the Hazelburn board as described in the following sections.

1.1.1 Select VDD_OTP from USB

To select the USB supply as the source for VDD_OTP, the P2 link must be configured to VDD_2V5. In this configuration, the OTP programming supply is enabled by setting the P3 link to VDD_VBUS.

The programming supply is disabled by default (P3 link selects GND).

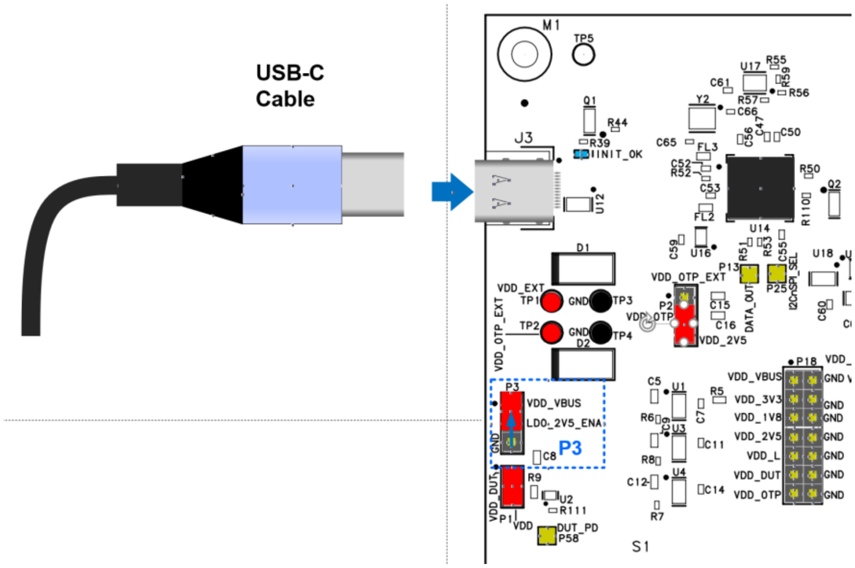


Figure 3 Programming Supply from USB

1.1.2 Select VDD_OTP from External Supply

To select an external supply as the source for VDD_OTP, the P2 link must be configured to VDD_OTP_EXT. In this configuration, an external source (2.5 V) can be used as the programming supply using test points TP2 (VDD_OTP_EXT) and TP4 (GND).

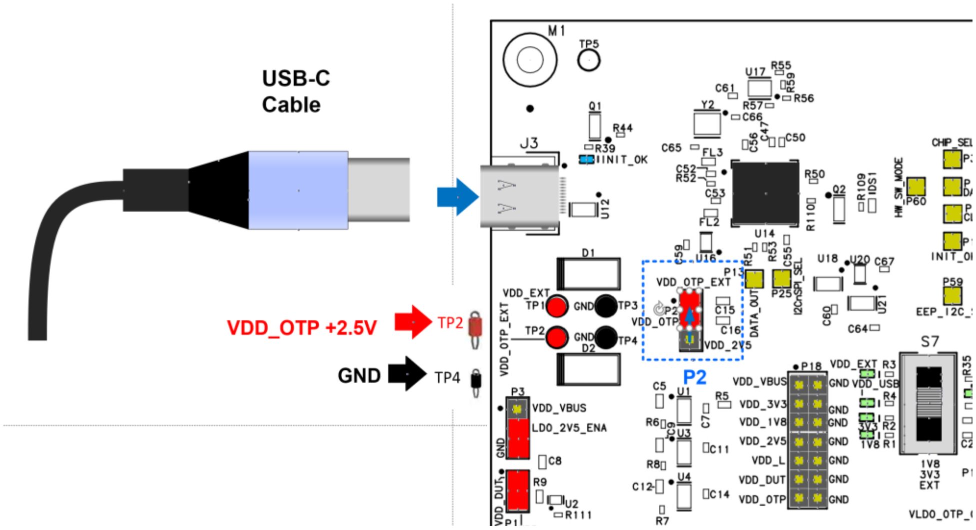


Figure 4 External Programming Supply

1.2 CDB2600-DC-SKT Daughter Card

The Hazelburn system uses interchangeable daughter cards to support a variety of devices. The CDB2600-DC-SKT is a daughter card designed to support OTP programming. The daughter card incorporates a socket for a CS2600 device.

The CDB2600-DC-SKT daughter card connects to J1 and J2 of the Hazelburn board as shown in Figure 5. The daughter-card connectors are keyed and can only plugged in one way.

Caution:

Daughter cards should not be inserted or removed while the Hazelburn system is powered or with external clock generators enabled. Fully disconnect or power down external power supply and disable or remove external clock generators before changing daughter cards.

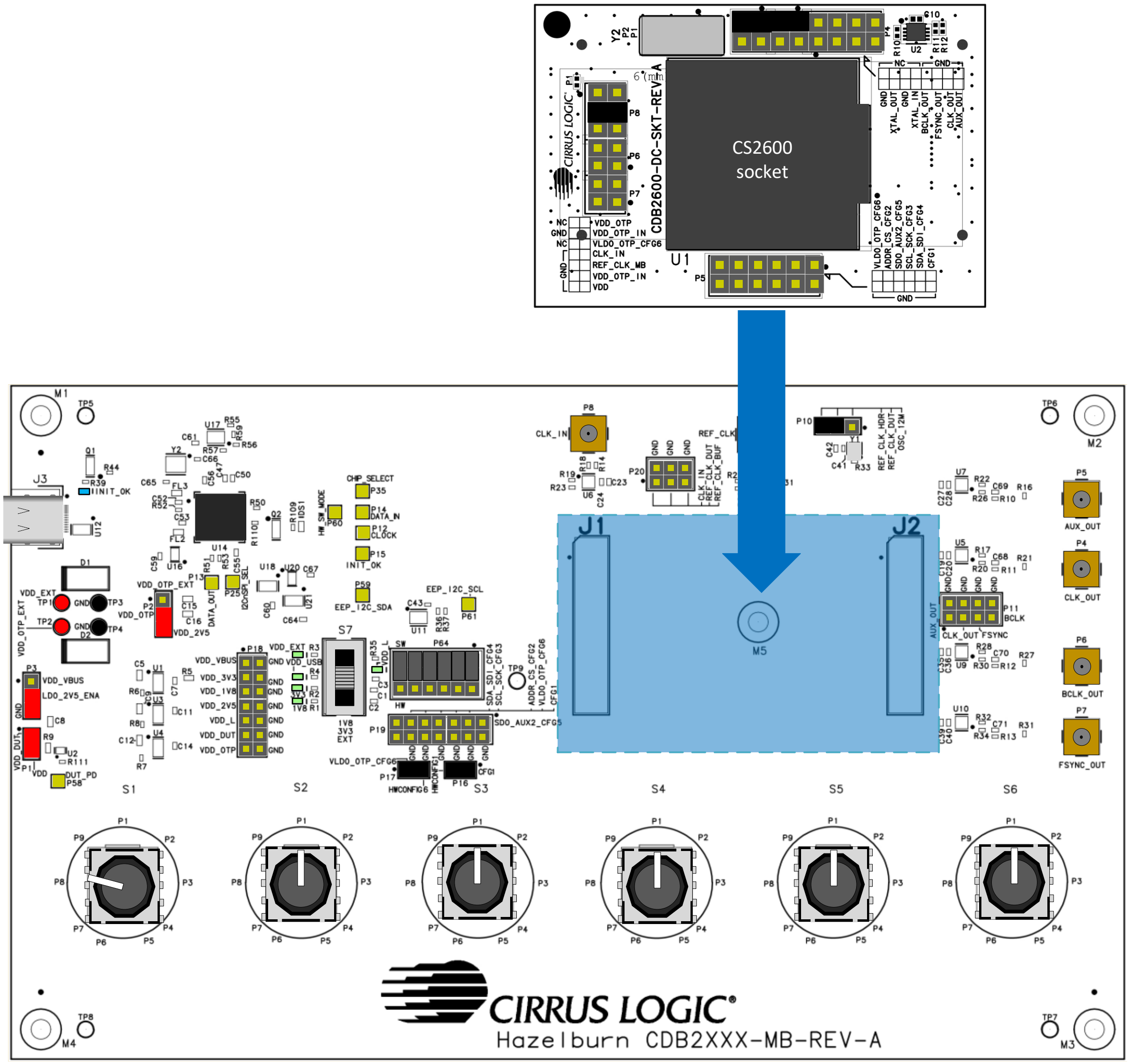


Figure 5 Hazelburn Daughter Card Connection

The top side of the CDB2600-DC-SKT daughter card is shown in Figure 6.

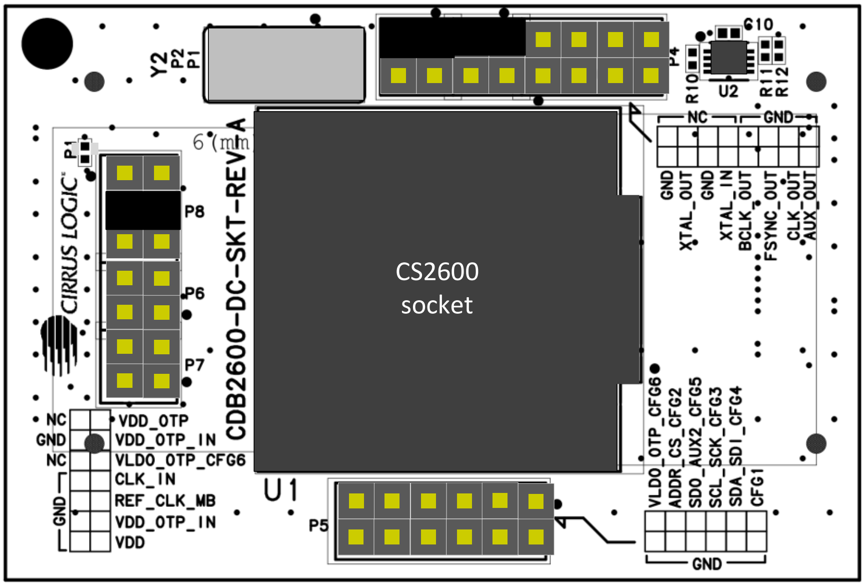


Figure 6 CDB2600-DC-SKT Daughter Card Top Side

The bottom side of the CDB2600-DC-SKT daughter card is shown in Figure 7.

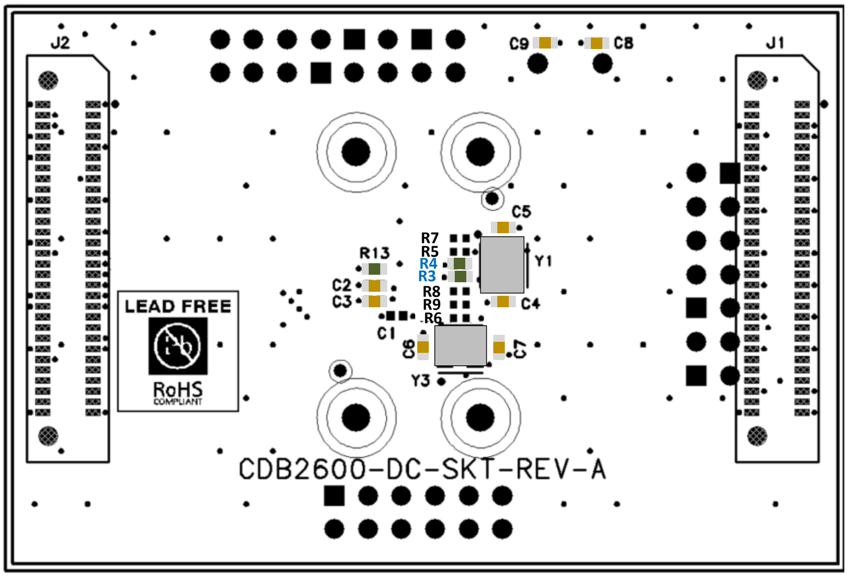


Figure 7 CDB2600-DC-SKT Daughter Card Bottom Side

Further information regarding the CDB2600-DC-SKT daughter card is provided in Section 4.

1.2.1 Select VDD_OTP from Internal Regulator (VDD = 3.3V)

To select the internal regulator as the source for VDD_OTP, the P8 link must be placed to connect Pin 4 (VDD_OTP_IN) and Pin 6 (VLDO_OTP). In this configuration, the output of the regulator, VLDO_OTP, is connected to VDD_OTP.

This configuration should only be used if the VDD supply is 3.3 V. See Section 1.1 to set the VDD operating voltage.

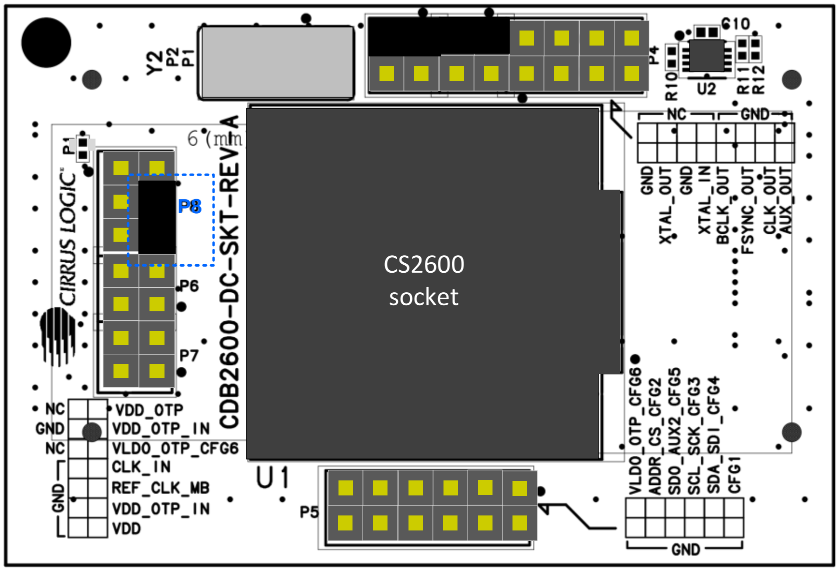


Figure 8 CDB2600-DC-SKT with VDD_OTP from Internal Regulator

1.2.2 Select VDD_OTP from Hazelburn System (VDD = 1.8V)

To select the Hazelburn system as the source for VDD_OTP, the P8 link must be placed to connect Pin 2 (VDD_OTP) and Pin 4 (VDD_OTP_IN). In this configuration, the 2.5 V programming supply is provided from the Hazelburn system as described in Section 1.1.

This configuration must be used if the VDD supply is 1.8 V. See Section 1.1 to set the VDD operating voltage.

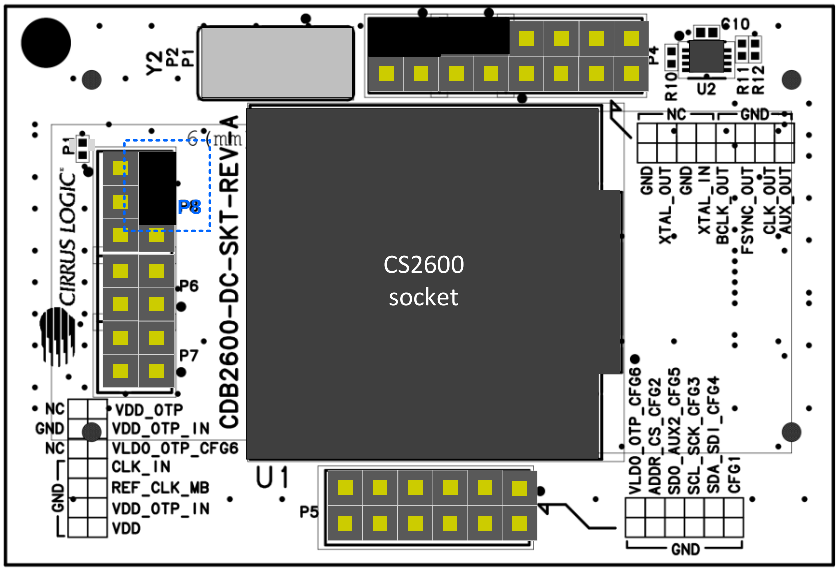
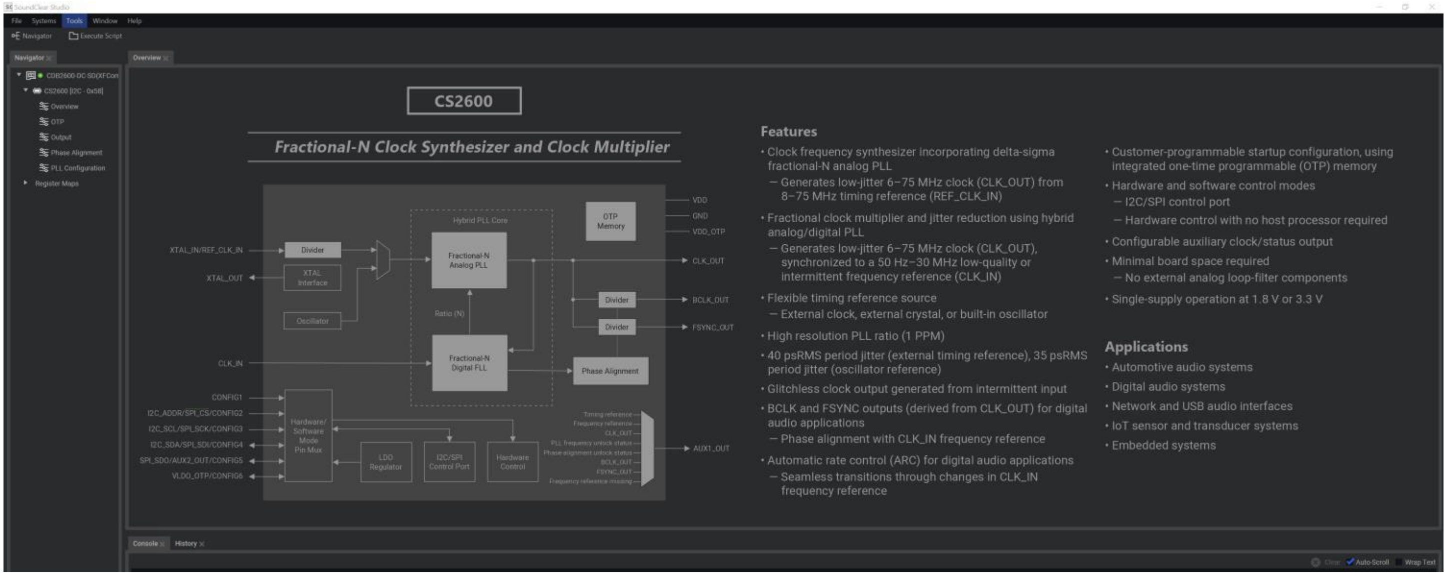


Figure 9 CDB2600-DC-SKT with VDD_OTP from Hazelburn System

2 SoundClear Studio Support

2.1 SoundClear Studio

SoundClear Studio (SCS) is a PC/Mac-based tool used to configure Cirrus Logic devices. The tools suite provides support for evaluation and development and can be used with Hazelburn system and associated daughter cards.



Features

- Clock frequency synthesizer incorporating delta-sigma fractional-N analog PLL
  - Generates low-jitter 6–75 MHz clock (CLK_OUT) from 8–75 MHz timing reference (REF_CLK_IN)
- Fractional clock multiplier and jitter reduction using hybrid analog/digital PLL
  - Generates low-jitter 6–75 MHz clock (CLK_OUT), synchronized to a 50 Hz–30 MHz low-quality or intermittent frequency reference (CLK_IN)
- Flexible timing reference source
  - External clock, external crystal, or built-in oscillator
- High resolution PLL ratio (1 PPM)
- 40 psRMS period jitter (external timing reference), 35 psRMS period jitter (oscillator reference)
- Glitchless clock output generated from intermittent input
- BCLK and FSYNC outputs (derived from CLK_OUT) for digital audio applications
  - Phase alignment with CLK_IN frequency reference
- Automatic rate control (ARC) for digital audio applications
  - Seamless transitions through changes in CLK_IN frequency reference
- Customer-programmable startup configuration, using integrated one-time programmable (OTP) memory
- Hardware and software control modes
  - I2C/SPI control port
  - Hardware control with no host processor required
- Configurable auxiliary clock/status output
- Minimal board space required
  - No external analog loop-filter components
- Single-supply operation at 1.8 V or 3.3 V

Applications

- Automotive audio systems
- Digital audio systems
- Network and USB audio interfaces
- IoT sensor and transducer systems
- Embedded systems

Figure 10 SoundClear Studio

The latest release of SoundClear Studio is available on the Cirrus Logic website and on the Cirrus Logic software (please contact your Cirrus Logic representative for access).

Note that, by downloading software from the Cirrus Logic website or software portal, you agree to the terms of our license agreement; please read the terms before downloading.

2.2 SoundClear Studio Quick Start Guide

2.2.1 Installing Packages

Each daughter card has its own individual SoundClear Studio package that must be installed separately from the main SoundClear Studio Software. These are installed from the main menu using **"File → Install Package..."**. Multiple packages can be installed together by selecting more than one using the file dialog.

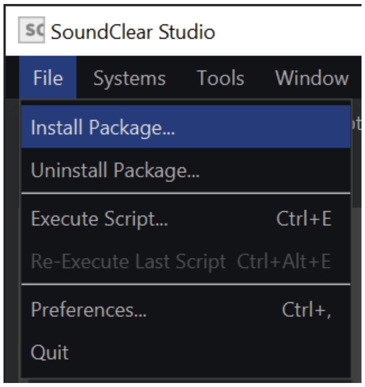


Figure 11 SoundClear Studio – Installing Board Packages

2.2.2 SoundClear Studio User Guide

The SoundClear Studio User Guide can be accessed from the main menu using **“Help → Open Help Contents..."**

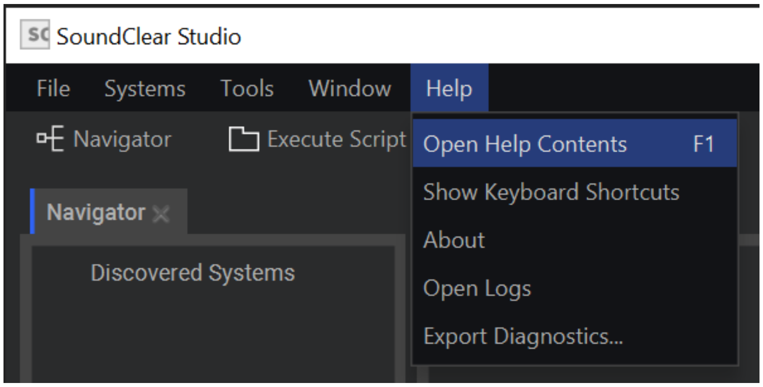


Figure 12 SoundClear Studio – User Guide

2.2.3 Executing SoundClear Studio Scripts

SoundClear Studio provides the ability to interact with the device register map using Python scripts. These scripts can sequence register operations to configure the device into desired states, which can then be executed from SoundClear Studio using **“File→Execute Script…”**

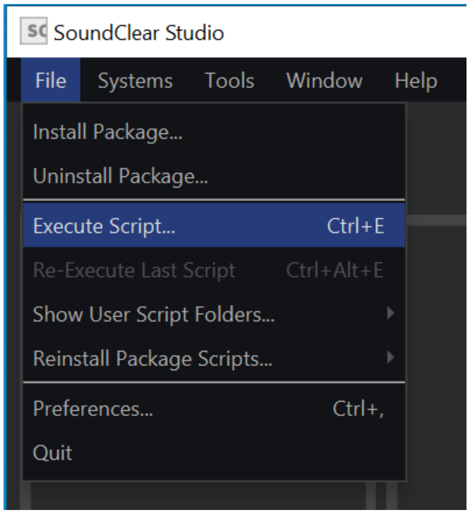


Figure 13 SoundClear Studio – Executing Script

The daughter-card SoundClear Studio package installs a set of scripts to configure the device for common use cases. The scripts are available at <User Documents>\Cirrus Logic\SCS\Scripts\<Package Name>.

The scripts can be accessed from SoundClear Studio using **“File→Show User Script Folder…→<Package Name>”**

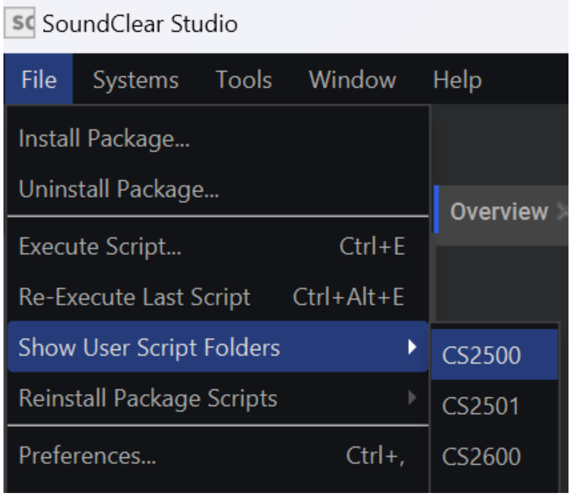


Figure 14 SoundClear Studio – Show User Script Folder

3 Customer OTP Programming

The CS2600 supports two different methods for OTP programming: prototyping and production respectively. A device may be programmed using either method, but not both.

3.1 OTP Programming Workflow

Prototype programming is used to fine-tune the device settings for a specific application. The prototype programming configures selected fields only, with all other fields initializing to their respective default values.

Prototype programming is configured using SoundClear Studio (SCS). The programmed contents of the OTP memory are loaded during startup. Note there is no error checking when using the prototype programming option.

Iterative programming is supported, with modified values defined for specific fields. If a field is programmed multiple times in the OTP memory, the most recent programmed value is applied during startup.

The OTP memory can be programmed multiple times. The programming limit depends on which parameters are being configured. As an example, however, it is possible to program and modify the PLL ratio more than 25 times.

Once a final prototype configuration is achieved, a production programming file (image.csv) can be exported for programming into production samples (in a production environment). See Section 3.3 for further details.

The diagram below shows the OTP programming workflow using both types of OTP programming processes in the two environments where they belong (development lab and production).

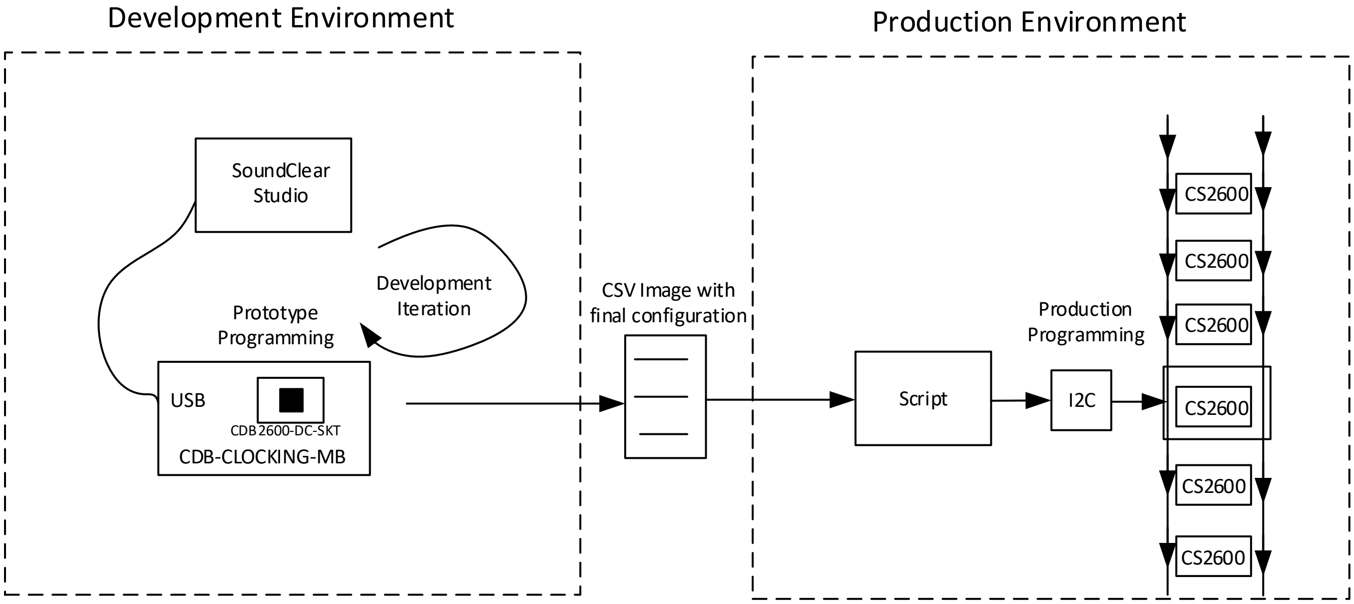


Figure 15 OTP Programming Workflow

3.2 Prototype OTP Programming

Prototype programming is supported using the Hazelburn system (CDB-CLOCKING-MB with CDB2600-DC-SKT daughter card), along with the SoundClear Studio (SCS) tool.

The OTP programming supply must be provided as described in Section 1.

Prior to programming the device, the required register settings must be identified for the target application. The register settings can be configured manually using SCS or by loading a script (as described in Section 2.2.3).

Once the register settings have been configured, the OTP panel should be opened using the Navigator pane on SCS, as shown in Figure 16.

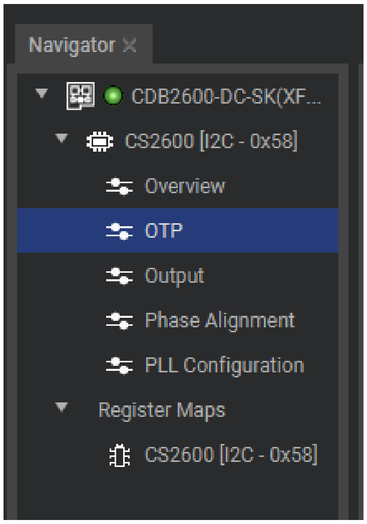
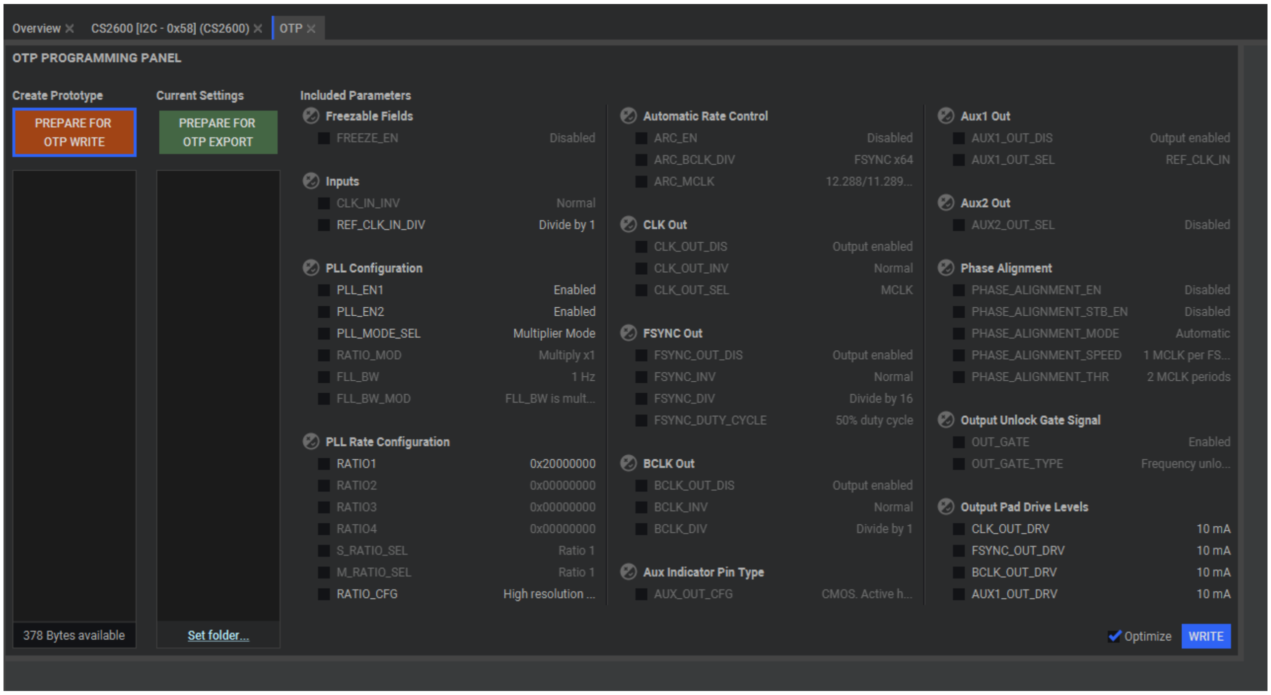


Figure 16 SoundClear Studio – Open OTP Panel

The OTP Programming Panel supports two options. The “Prepare for OTP Write” selection is used to prepare the device for prototype programming. The “Prepare for OTP Export” selection is used to create a CSV file for production programming.

For prototype programming, select “Prepare for OTP Write”. A list of all registers is displayed; fields whose values have changed from default are highlighted.

The preparation for OTP Write is shown in Figure 17. Note that, if the device has already been programmed as a production device, an error message indicates that prototype programming is no longer possible.



Freezable Fields

□ FREEZE_EN	Disabled

Inputs

□ CLK_IN_INV	Normal
□ REF_CLK_IN_DIV	Divide by 1

PLL Configuration

□ PLL_EN1	Enabled
□ PLL_EN2	Enabled
□ PLL_MODE_SEL	Multiplier Mode
□ RATIO_MOD	Multiply x1
□ FLL_BW	1 Hz
□ FLL_BW_MOD	FLL_BW is mult...

PLL Rate Configuration

□ RATIO1	0x20000000
□ RATIO2	0x00000000
□ RATIO3	0x00000000
□ RATIO4	0x00000000
□ S_RATIO_SEL	Ratio 1
□ M_RATIO_SEL	Ratio 1
□ RATIO_CFG	High resolution ...

Automatic Rate Control

□ ARC_EN	Disabled
□ ARC_BCLK_DIV	FSYNC x64
□ ARC_MCLK	12.288/11.289...

CLK Out

□ CLK_OUT_DIS	Output enabled
□ CLK_OUT_INV	Normal
□ CLK_OUT_SEL	MCLK

FSYNC Out

□ FSYNC_OUT_DIS	Output enabled
□ FSYNC_INV	Normal
□ FSYNC_DIV	Divide by 16
□ FSYNC_DUTY_CYCLE	50% duty cycle

BCLK Out

□ BCLK_OUT_DIS	Output enabled
□ BCLK_INV	Normal
□ BCLK_DIV	Divide by 1

Aux Indicator Pin Type

□ AUX_OUT_CFG	CMOS. Active h...

Aux1 Out

□ AUX1_OUT_DIS	Output enabled
□ AUX1_OUT_SEL	REF_CLK_IN

Aux2 Out

□ AUX2_OUT_SEL	Disabled

Phase Alignment

□ PHASE_ALIGNMENT_EN	Disabled
□ PHASE_ALIGNMENT_STB_EN	Disabled
□ PHASE_ALIGNMENT_MODE	Automatic
□ PHASE_ALIGNMENT_SPEED	1 MCLK per FS...
□ PHASE_ALIGNMENT_THR	2 MCLK periods

Output Unlock Gate Signal

□ OUT_GATE	Enabled
□ OUT_GATE_TYPE	Frequency unlo...

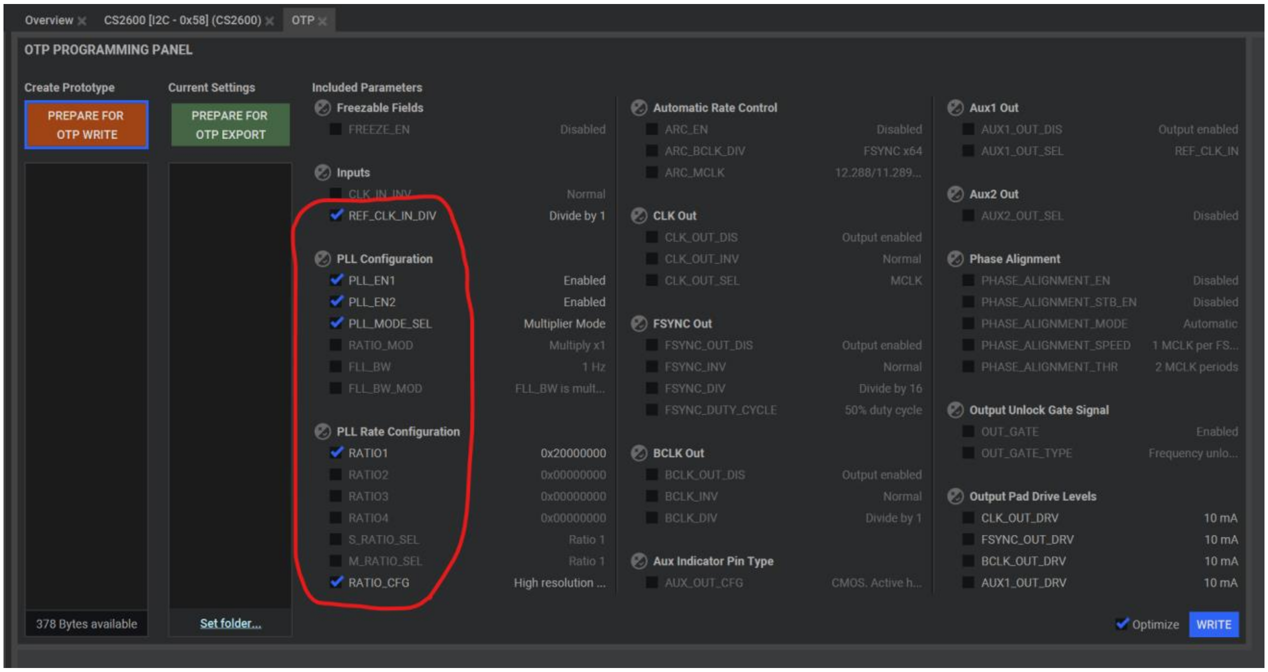
Output Pad Drive Levels

□ CLK_OUT_DRV	10 mA
□ FSYNC_OUT_DRV	10 mA
□ BCLK_OUT_DRV	10 mA
□ AUX1_OUT_DRV	10 mA

☑ Optimize

Figure 17 OTP panel – Prepare for OTP Write

Use the check boxes to select the register fields for prototype programming. For typical use cases, all of the fields with non-default values should be selected.



378 Bytes available

Included Parameters

Freezable Fields
- □ FREEZE_EN Disabled

Inputs
- □ CLK_IN_INV Normal
- ☑ REF_CLK_IN_DIV Divide by 1

PLL Configuration
- ☑ PLL_EN1 Enabled
- ☑ PLL_EN2 Enabled
- ☑ PLL_MODE_SEL Multiplier Mode
- □ RATIO_MOD Multiply x1
- □ FLL_BW 1 Hz
- □ FLL_BW_MOD FLL_BW is mult...

PLL Rate Configuration
- ☑ RATIO1 0x20000000
- □ RATIO2 0x00000000
- □ RATIO3 0x00000000
- □ RATIO4 0x00000000
- □ S_RATIO_SEL Ratio 1
- □ M_RATIO_SEL Ratio 1
- ☑ RATIO_CFG High resolution ...

Automatic Rate Control
- □ ARC_EN Disabled
- □ ARC_BCLK_DIV FSYNC x64
- □ ARC_MCLK 12.288/11.289...

CLK Out
- □ CLK_OUT_DIS Output enabled
- □ CLK_OUT_INV Normal
- □ CLK_OUT_SEL MCLK

FSYNC Out
- □ FSYNC_OUT_DIS Output enabled
- □ FSYNC_INV Normal
- □ FSYNC_DIV Divide by 16
- □ FSYNC_DUTY_CYCLE 50% duty cycle

BCLK Out
- □ BCLK_OUT_DIS Output enabled
- □ BCLK_INV Normal
- □ BCLK_DIV Divide by 1

Aux Indicator Pin Type
- □ AUX_OUT_CFG CMOS. Active h...

Aux1 Out
- □ AUX1_OUT_DIS Output enabled
- □ AUX1_OUT_SEL REF_CLK_IN

Aux2 Out
- □ AUX2_OUT_SEL Disabled

Phase Alignment
- □ PHASE_ALIGNMENT_EN Disabled
- □ PHASE_ALIGNMENT_STB_EN Disabled
- □ PHASE_ALIGNMENT_MODE Automatic
- □ PHASE_ALIGNMENT_SPEED 1 MCLK per FS...
- □ PHASE_ALIGNMENT_THR 2 MCLK periods

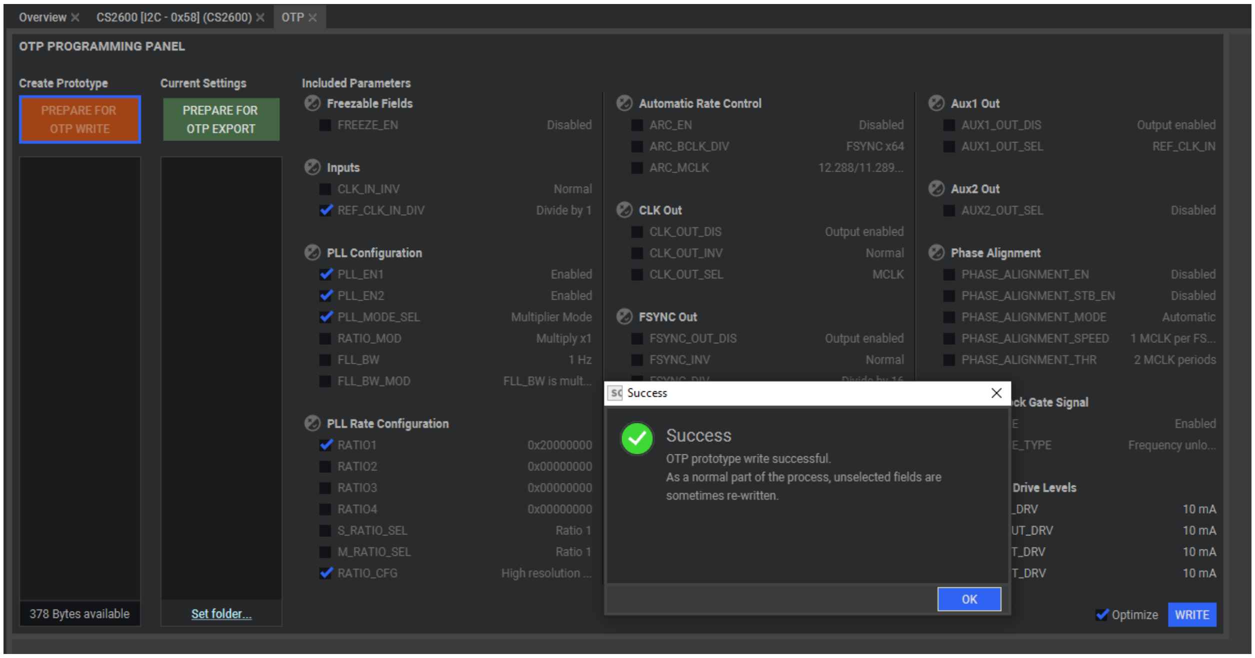
Output Unlock Gate Signal
- □ OUT_GATE Enabled
- □ OUT_GATE_TYPE Frequency unlo...

Output Pad Drive Levels
- □ CLK_OUT_DRV 10 mA
- □ FSYNC_OUT_DRV 10 mA
- □ BCLK_OUT_DRV 10 mA
- □ AUX1_OUT_DRV 10 mA

☑ Optimize

Figure 18 OTP Panel – Select Register Fields

The prototype programming is executed by clicking on “Write”. A successful write is indicated as shown in Figure 19.



378 Bytes available

Success
OTP prototype write successful.
As a normal part of the process, unselected fields are sometimes re-written.

Included Parameters

Freezable Fields
- □ FREEZE_EN Disabled

Inputs
- □ CLK_IN_INV Normal
- ☑ REF_CLK_IN_DIV Divide by 1

PLL Configuration
- ☑ PLL_EN1 Enabled
- ☑ PLL_EN2 Enabled
- ☑ PLL_MODE_SEL Multiplier Mode
- □ RATIO_MOD Multiply x1
- □ FLL_BW 1 Hz
- □ FLL_BW_MOD FLL_BW is mult...

PLL Rate Configuration
- ☑ RATIO1 0x20000000
- □ RATIO2 0x00000000
- □ RATIO3 0x00000000
- □ RATIO4 0x00000000
- □ S_RATIO_SEL Ratio 1
- □ M_RATIO_SEL Ratio 1
- ☑ RATIO_CFG High resolution ...

Automatic Rate Control
- □ ARC_EN Disabled
- □ ARC_BCLK_DIV FSYNC x64
- □ ARC_MCLK 12.288/11.289...

CLK Out
- □ CLK_OUT_DIS Output enabled
- □ CLK_OUT_INV Normal
- □ CLK_OUT_SEL MCLK

FSYNC Out
- □ FSYNC_OUT_DIS Output enabled
- □ FSYNC_INV Normal

Aux1 Out
- □ AUX1_OUT_DIS Output enabled
- □ AUX1_OUT_SEL REF_CLK_IN

Aux2 Out
- □ AUX2_OUT_SEL Disabled

Phase Alignment
- □ PHASE_ALIGNMENT_EN Disabled
- □ PHASE_ALIGNMENT_STB_EN Disabled
- □ PHASE_ALIGNMENT_MODE Automatic
- □ PHASE_ALIGNMENT_SPEED 1 MCLK per FS...
- □ PHASE_ALIGNMENT_THR 2 MCLK periods

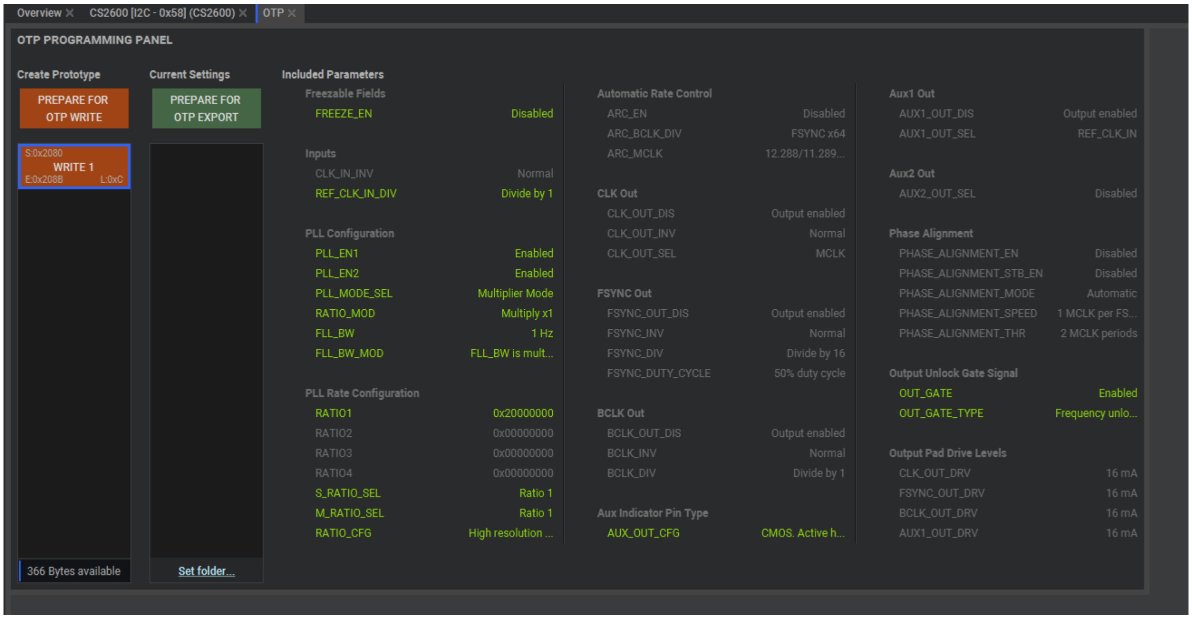
ck Gate Signal
- E Enabled
- E_TYPE Frequency unlo...

Drive Levels
- _DRV 10 mA
- UT_DRV 10 mA
- T_DRV 10 mA
- T_DRV 10 mA

☑ Optimize

Figure 19 Successful OTP Write

Once the write is completed, the OTP Programming Panel is updated to show the number of prototype writes performed and the number of OTP bytes left available for programming. Fields whose values have changed from default are highlighted. Note that unselected fields are sometimes written, in order to optimize the OTP memory.



Freezable Fields

FREEZE_EN	Disabled

Inputs

CLK_IN_INV	Normal
REF_CLK_IN_DIV	Divide by 1

PLL Configuration

PLL_EN1	Enabled
PLL_EN2	Enabled
PLL_MODE_SEL	Multiplier Mode
RATIO_MOD	Multiply x1
FLL_BW	1 Hz
FLL_BW_MOD	FLL_BW is mult...

PLL Rate Configuration

RATIO1	0x20000000
RATIO2	0x00000000
RATIO3	0x00000000
RATIO4	0x00000000
S_RATIO_SEL	Ratio 1
M_RATIO_SEL	Ratio 1
RATIO_CFG	High resolution ...

Automatic Rate Control

ARC_EN	Disabled
ARC_BCLK_DIV	FSYNC x64
ARC_MCLK	12.288/11.289...

CLK Out

CLK_OUT_DIS	Output enabled
CLK_OUT_INV	Normal
CLK_OUT_SEL	MCLK

FSYNC Out

FSYNC_OUT_DIS	Output enabled
FSYNC_INV	Normal
FSYNC_DIV	Divide by 16
FSYNC_DUTY_CYCLE	50% duty cycle

BCLK Out

BCLK_OUT_DIS	Output enabled
BCLK_INV	Normal
BCLK_DIV	Divide by 1

Aux Indicator Pin Type

AUX_OUT_CFG	CMOS. Active h...

Aux1 Out

AUX1_OUT_DIS	Output enabled
AUX1_OUT_SEL	REF_CLK_IN

Aux2 Out

AUX2_OUT_SEL	Disabled

Phase Alignment

PHASE_ALIGNMENT_EN	Disabled
PHASE_ALIGNMENT_STB_EN	Disabled
PHASE_ALIGNMENT_MODE	Automatic
PHASE_ALIGNMENT_SPEED	1 MCLK per FS...
PHASE_ALIGNMENT_THR	2 MCLK periods

Output Unlock Gate Signal

OUT_GATE	Enabled
OUT_GATE_TYPE	Frequency unlo...

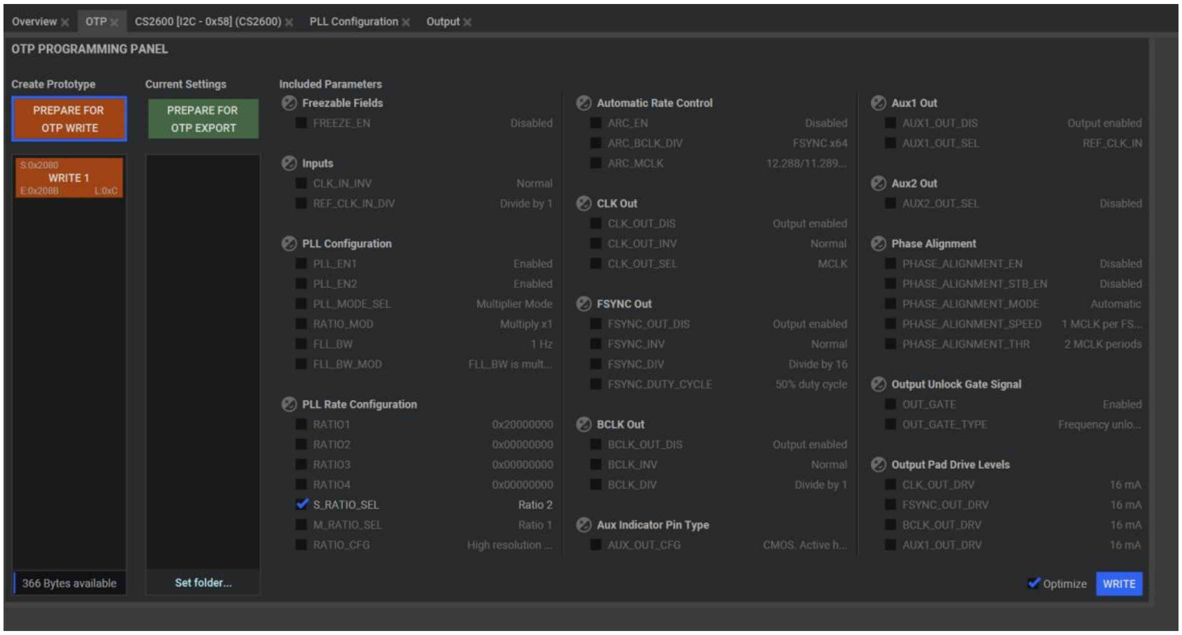
Output Pad Drive Levels

CLK_OUT_DRV	16 mA
FSYNC_OUT_DRV	16 mA
BCLK_OUT_DRV	16 mA
AUX1_OUT_DRV	16 mA

Figure 20 SoundClear Studio – Updated OTP Panel

The prototype programming can be performed multiple times, to further modify the register settings. The programming operations are cumulative – any fields that were configured by Write 1 may be changed again by Write 2, and so on.

After updating the register fields (e.g., using the PLL Configuration Panel), select “Prepare for OTP Write” to configure an update to the OTP prototype program. Use the check boxes to select the fields for programming. In the example shown in Figure 21, the S_RATIO_SEL field is selected for update.



Freezable Fields

☐	FREEZE_EN	Disabled

Inputs

☐	CLK_IN_INV	Normal
☐	REF_CLK_IN_DIV	Divide by 1

PLL Configuration

☐	PLL_EN1	Enabled
☐	PLL_EN2	Enabled
☐	PLL_MODE_SEL	Multiplier Mode
☐	RATIO_MOD	Multiply x1
☐	FLL_BW	1 Hz
☐	FLL_BW_MOD	FLL_BW is mult...

PLL Rate Configuration

☐	RATIO1	0x20000000
☐	RATIO2	0x00000000
☐	RATIO3	0x00000000
☐	RATIO4	0x00000000
☑	S_RATIO_SEL	Ratio 2
☐	M_RATIO_SEL	Ratio 1
☐	RATIO_CFG	High resolution ...

Automatic Rate Control

☐	ARC_EN	Disabled
☐	ARC_BCLK_DIV	FSYNC x64
☐	ARC_MCLK	12.288/11.289...

CLK Out

☐	CLK_OUT_DIS	Output enabled
☐	CLK_OUT_INV	Normal
☐	CLK_OUT_SEL	MCLK

FSYNC Out

☐	FSYNC_OUT_DIS	Output enabled
☐	FSYNC_INV	Normal
☐	FSYNC_DIV	Divide by 16
☐	FSYNC_DUTY_CYCLE	50% duty cycle

BCLK Out

☐	BCLK_OUT_DIS	Output enabled
☐	BCLK_INV	Normal
☐	BCLK_DIV	Divide by 1

Aux Indicator Pin Type

☐	AUX_OUT_CFG	CMOS. Active h...

Aux1 Out

☐	AUX1_OUT_DIS	Output enabled
☐	AUX1_OUT_SEL	REF_CLK_IN

Aux2 Out

☐	AUX2_OUT_SEL	Disabled

Phase Alignment

☐	PHASE_ALIGNMENT_EN	Disabled
☐	PHASE_ALIGNMENT_STB_EN	Disabled
☐	PHASE_ALIGNMENT_MODE	Automatic
☐	PHASE_ALIGNMENT_SPEED	1 MCLK per FS...
☐	PHASE_ALIGNMENT_THR	2 MCLK periods

Output Unlock Gate Signal

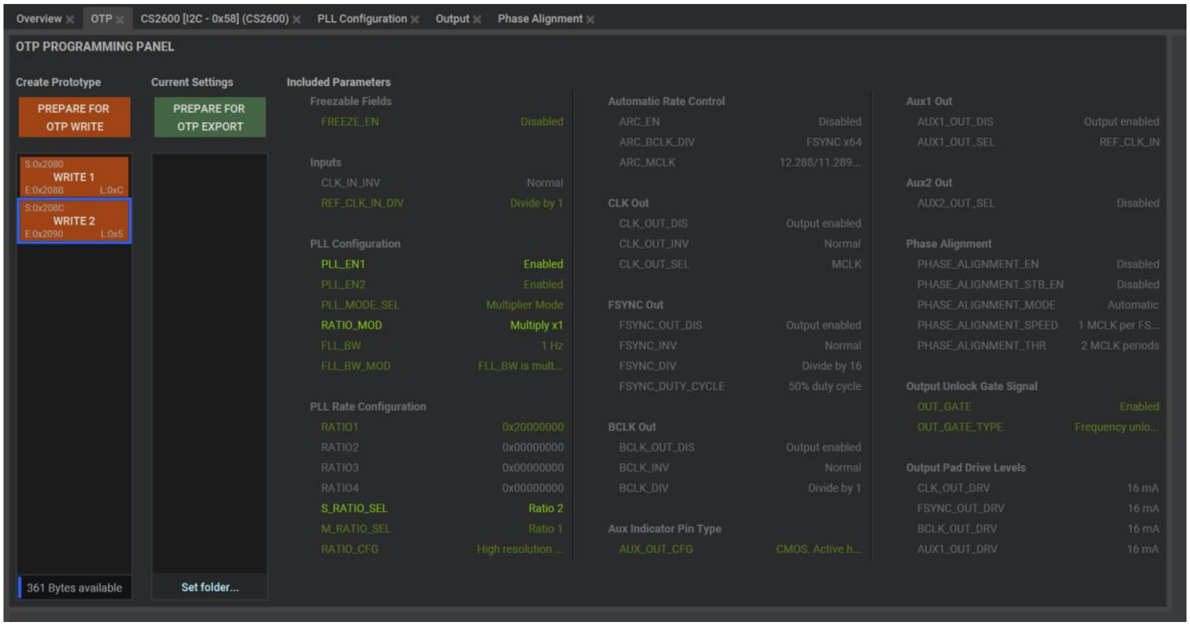
☐	OUT_GATE	Enabled
☐	OUT_GATE_TYPE	Frequency unlo...

Output Pad Drive Levels

☐	CLK_OUT_DRV	16 mA
☐	FSYNC_OUT_DRV	16 mA
☐	BCLK_OUT_DRV	16 mA
☐	AUX1_OUT_DRV	16 mA

Figure 21 SoundClear Studio – Updated OTP Panel with S_RATIO_SEL Selected

After a successful write, the OTP Programming Panel is updated again to show the number of prototype writes performed. In Figure 20, the update (Write 2) has been added.



Freezable Fields

FREEZE_EN	Disabled

Inputs

CLK_IN_INV	Normal
REF_CLK_IN_DIV	Divide by 1

PLL Configuration

PLL_EN1	Enabled
PLL_EN2	Enabled
PLL_MODE_SEL	Multiplier Mode
RATIO_MOD	Multiply x1
FLL_BW	1 Hz
FLL_BW_MOD	FLL_BW is mult...

PLL Rate Configuration

RATIO1	0x20000000
RATIO2	0x00000000
RATIO3	0x00000000
RATIO4	0x00000000
S_RATIO_SEL	Ratio 2
M_RATIO_SEL	Ratio 1
RATIO_CFG	High resolution ...

Automatic Rate Control

ARC_EN	Disabled
ARC_BCLK_DIV	FSYNC x64
ARC_MCLK	12.288/11.289...

CLK Out

CLK_OUT_DIS	Output enabled
CLK_OUT_INV	Normal
CLK_OUT_SEL	MCLK

FSYNC Out

FSYNC_OUT_DIS	Output enabled
FSYNC_INV	Normal
FSYNC_DIV	Divide by 16
FSYNC_DUTY_CYCLE	50% duty cycle

BCLK Out

BCLK_OUT_DIS	Output enabled
BCLK_INV	Normal
BCLK_DIV	Divide by 1

Aux Indicator Pin Type

AUX_OUT_CFG	CMOS. Active h...

Aux1 Out

AUX1_OUT_DIS	Output enabled
AUX1_OUT_SEL	REF_CLK_IN

Aux2 Out

AUX2_OUT_SEL	Disabled

Phase Alignment

PHASE_ALIGNMENT_EN	Disabled
PHASE_ALIGNMENT_STB_EN	Disabled
PHASE_ALIGNMENT_MODE	Automatic
PHASE_ALIGNMENT_SPEED	1 MCLK per FS...
PHASE_ALIGNMENT_THR	2 MCLK periods

Output Unlock Gate Signal

OUT_GATE	Enabled
OUT_GATE_TYPE	Frequency unlo...

Output Pad Drive Levels

CLK_OUT_DRV	16 mA
FSYNC_OUT_DRV	16 mA
BCLK_OUT_DRV	16 mA
AUX1_OUT_DRV	16 mA

Figure 22 SoundClear Studio – Updated OTP Panel after ‘Write’ has Selected and Completed

3.3 Production OTP Programming

Production programming is implemented using an image that defines the required settings of all the configurable registers. The image is generated using SoundClear Studio (SCS).

The image includes a user-selectable ID field (1–7) and an error correction code (ECC) field. The ECC supports 3-bit error detection and 2-bit error correction.

The following three items are required for production programming of the OTP memory:

- A CSV image file (image.csv), generated in SCS following the prototyping stage.
- A custom code script to parse the image file and execute the required I2C/SPI writes to the chip.
- The OTP programming supply must be provided to VDD_OTP. See Section 1.

The programmed contents of the OTP memory are loaded during startup. If the OTP memory contains an uncorrectable error, the clock outputs are disabled, and the device startup is aborted. An OTP error is indicated using ERR_STS7.

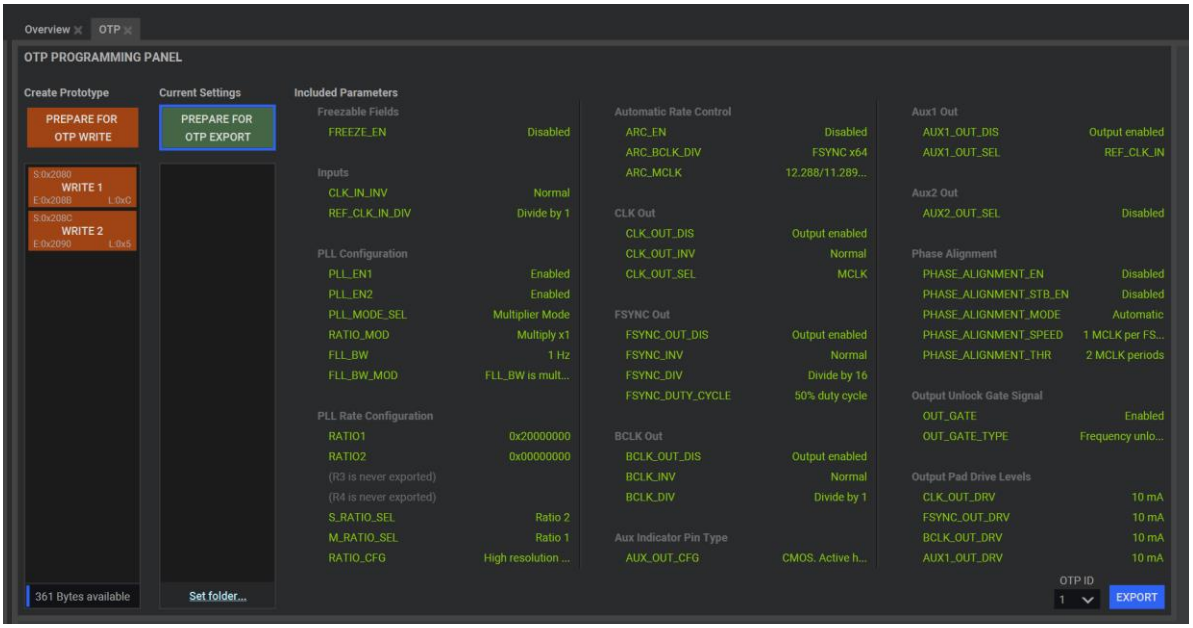
The number of errors detected in the OTP memory is indicated using OTP_IMG_ECC_STS. If errors are present after production programming, it is recommended to reprogram the device (even if the errors are correctable). Eliminating any errors maximizes the long-term reliability of the OTP program in the target application.

The OTP memory can be programmed up to seven times. Each time the OTP memory is programmed, the previous images are automatically superseded.

3.3.1 Generating CSV Image File in OTP Panel (SCS)

The CSV image file is generated using the “Prepare for OTP Export” option in the OTP programming panel as shown in Figure 23. The image ID is selectable in the range 1–7.

The .CSV export is created by clicking on “Export”.



Included Parameters

Freezable Fields	
FREEZE_EN	Disabled

Inputs	
CLK_IN_INV	Normal
REF_CLK_IN_DIV	Divide by 1

PLL Configuration	
PLL_EN1	Enabled
PLL_EN2	Enabled
PLL_MODE_SEL	Multiplier Mode
RATIO_MOD	Multiply x1
FLL_BW	1 Hz
FLL_BW_MOD	FLL_BW is mult...

PLL Rate Configuration	
RATIO1	0x20000000
RATIO2	0x00000000
(R3 is never exported)	
(R4 is never exported)	
S_RATIO_SEL	Ratio 2
M_RATIO_SEL	Ratio 1
RATIO_CFG	High resolution ...

Automatic Rate Control	
ARC_EN	Disabled
ARC_BCLK_DIV	FSYNC x64
ARC_MCLK	12.288/11.289...

CLK Out	
CLK_OUT_DIS	Output enabled
CLK_OUT_INV	Normal
CLK_OUT_SEL	MCLK

FSYNC Out	
FSYNC_OUT_DIS	Output enabled
FSYNC_INV	Normal
FSYNC_DIV	Divide by 16
FSYNC_DUTY_CYCLE	50% duty cycle

BCLK Out	
BCLK_OUT_DIS	Output enabled
BCLK_INV	Normal
BCLK_DIV	Divide by 1

Aux Indicator Pin Type	
AUX_OUT_CFG	CMOS. Active h...

Aux1 Out	
AUX1_OUT_DIS	Output enabled
AUX1_OUT_SEL	REF_CLK_IN

Aux2 Out	
AUX2_OUT_SEL	Disabled

Phase Alignment	
PHASE_ALIGNMENT_EN	Disabled
PHASE_ALIGNMENT_STB_EN	Disabled
PHASE_ALIGNMENT_MODE	Automatic
PHASE_ALIGNMENT_SPEED	1 MCLK per FS...
PHASE_ALIGNMENT_THR	2 MCLK periods

Output Unlock Gate Signal	
OUT_GATE	Enabled
OUT_GATE_TYPE	Frequency unlo...

Output Pad Drive Levels	
CLK_OUT_DRV	10 mA
FSYNC_OUT_DRV	10 mA
BCLK_OUT_DRV	10 mA
AUX1_OUT_DRV	10 mA

OTP ID: 1

Figure 23 SoundClear Studio – Select Export in OTP Panel

3.3.2 Pseudo-Code Script Example

The CSV image contains a list of register writes, which must be parsed and written to the device for programming. The following pseudo-code script is provided for reference; it shows the sequences which should be executed in a production environment.

Note the writes to OTP_LDO_EN and OTP_LDO_DISCH are only required if the internal LDO regulator is used to provide the OTP programming supply (Option A in Section 1).

Stage 1 - Initialization and setup

# Software reset. Write 0x5A to execute a software reset.

```
writeReg(“SW_RESET”, 0x005A)

wait 200 us
```

After SW reset, read the status of OTP_MODE. If the status is 0 (Production Mode), the device has not been used for prototyping already. If the status is 1 (Prototyping Mode), the device has been used for prototyping mode and cannot be used for production programming.

# Check OTP_MODE status.

```
otp_sts = readReg(“OTP_STS6”, bit 0”)

if (otp_sts & 1)

{
      print ("Error: Device already programmed in prototyping mode!")
      exit()
}
```

Next, read the number of Production Mode images in OTP_IMG_NUM. If the status is 0x0, it indicates no image present. If the status is 0x1 to 0x7, it represents the number of images written. If the status is 0x7, the production OTP space is full.

# Check OTP_IMG_NUM status.

```
    img_num = readReg(“OTP_STS6”, bits 4,5,6”)
    If (img_num >6 )
    {
        print ("Error: OTP memory is full.")
        exit()
    }
```

# Unlock User Key control – it requires two writes to set (unlock): 0xAA followed by 0x55 sets the key.

```
    writeReg(“USER_KEY_REG”, 0x00AA)
    writeReg(“USER_KEY_REG”, 0x0055)
```

If the internal programming supply is used, disable the LDO regulator discharge path using OTP_LDO_DISCH.

# Set OTP_LDO_DISCH_EN to 0 (Disabled)

```
    writeReg(“OTP_VDD_CTRL”, 0x0000)
    wait 500 us
```

Configure the OTP programming supply.

# Set OTP_VDD_EN to 1 (Enabled)

```
    writeReg(“OTP_VDD_CTRL”, 0x0008)
    wait 100 us
```

If the internal programming supply is used, enable the internal LDO regulator using OTP_LDO_EN. If the internal regulator is not used, provide the external OTP programming supply to VDD_OTP.

# Set OTP_LDO_EN to 1 (Enabled). Note – for internal programming supply only. Omit this if an external OTP programming supply is used.

```
    writeReg(“OTP_VDD_CTRL”, 0x0009)
    wait 500 us
```

Stage 2 - Programming process

# Read in production csv file

```
    open('name.csv', mode='r') as csv_file:
    csv_reader = csv.DictReader(csv_file)
```

# Enable OTP programming – Set OTP_PROG_EN to 1 (OTP writes permissible)

```
    writeReg(“OTP_CONTROL1”, 0x041A)
```

************* Programming ******************

# Parse the CSV image file and write to OTP space. The CSV image file contains a header followed by 24 lines of address and data (value) pairs. The OTP data is written at register address 0x2340 – 0x236E.

CSV file example:

address	value
0x2340	0x1732
0x2342	0x0B08
0x2344	0x0000
0x2346	0x0000
0x2348	0x0100
0x234A	0x0109
0x234C	0x0810
0x234E	0x0000
0x2350	0x0020
0x2352	0x0000
0x2354	0x0000
0x2356	0x0000
0x2358	0x0000
0x235A	0x4444
0x235C	0x0000
0x235E	0x0000
0x2360	0x0000
0x2362	0x0000
0x2364	0x0000
0x2366	0x0000
0x2368	0x0000
0x236A	0xBB00
0x236C	0xC53B
0x236E	0x0FD2

```
line_count = 0
for row in csv_reader:
      if line_count == 0:
            print(f'Column names are {", ".join(row)}')
            line_count += 1
      print(f'\t{row["address"]} , {row["value"]}')
      writeReg(int(row["address"],16), int(row["value"],16))
      wait 200 us
line_count += 1
print(f'Processed {line_count} lines.')
```

***************Programming termination ********************

# Disable OTP programming – Set OTP_PROG_EN to 0 (OTP writes ignored)

```
writeReg(“OTP_CONTROL1”, 0x0418)
```

If the internal programming supply is used, disable the internal LDO regulator using OTP_LDO_EN. If the internal regulator is not used, disable the external OTP programming supply to VDD_OTP.

# Set OTP_LDO_EN to 0 (Disabled)

```
writeReg(“OTP_VDD_CTRL”, 0x0008)

wait 100 us
```

Disable the OTP programming supply.

# Set OTP_VDD_EN to 0 (Disabled)

```
writeReg(“OTP_VDD_CTRL”, 0x0000)

wait 500 us
```

If the internal programming supply is used, the LDO regulator output discharge path should be momentarily enabled for a period of at least two seconds.

# Set OTP_LDO_DISCH_ EN to 1 (Enabled)

```
writeReg(“OTP_VDD_CTRL”, 0x0002)

wait 2000 us
```

# Set OTP_LDO_DISCH_ EN to 0 (Disabled)

```
writeReg(“OTP_VDD_CTRL”, 0x0000)

wait 100 us
```

# Lock User key

```
writeReg(“USER_KEY_REG”, 0x0000)
```

Stage 3. Verification of image file settings

# Software reset. Write 0x5A to execute a software reset.

```
writeReg(“SW_RESET”, 0x005A)

wait 500 us
```

# Read back from register addresses 0x2300 to 0x232E and verify with image file register addresses 0x2340 to 0x236E for each of the 24 address/data pairs:

```
value = readReg(address)

      if (value != data)

{

      print (“Error: data mismatch at ‘address’. Read ‘value’, expected ‘data’.”)

      exit()

}
```

4 CDB2600-DC-SKT Reference Clock Selection

The CS2600 frequency reference (REF_CLK) can be provided using a crystal, external reference, or internal oscillator. This section describes how to configure the reference clock on the CDB2600-DC-SKT daughter card.

Note that the choice of reference has no effect on OTP programming. The REF_CLK options are documented here as the CDB2600-DC-SKT board is not described in the Hazelburn user guide.

The frequency reference (REF_CLK) is provided by a 12 MHz crystal (Y1) on the daughter card by default. Alternative sources for REF_CLK are configured using 0 Ω resistor links and PCB jumper connections.

The daughter card configuration for each option is described as follows.

- The 12 MHz crystal Y1 on the daughter card is the default REF_CLK source. The part number of the crystal is *NX3225SA-12.000M-STD-CRS-2* (SMD/SMT crystal); further information can be found in the crystal datasheet.

  The daughter card configuration (default) is R3, R4 populated (0 Ω); R5, R6, R7, R8, R9 non-populated.

  The configuration is illustrated in Figure 24.

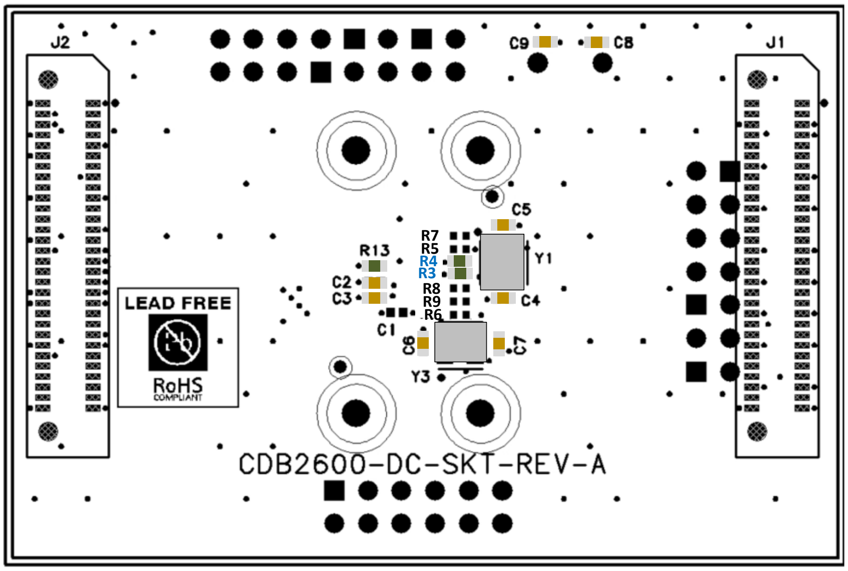


Figure 24 Daughter Card with Crystal Y1 Reference Clock Option (R3 and R4)

- The REF_CLK can be provided from the Hazelburn motherboard (CDB-CLOCKING-MB). This can be either an external signal via connector P9, or the 12 MHz crystal oscillator on the motherboard. See Section 1.3.1.1 of the Hazelburn user guide (CDB-CLOCKING_DS1386DB1.pdf) to configure the motherboard for the required option.

  The daughter card configuration is R9 populated (0 Ω); R3, R4, R5, R6, R7, R8 non-populated.

  The configuration is illustrated in Figure 25.

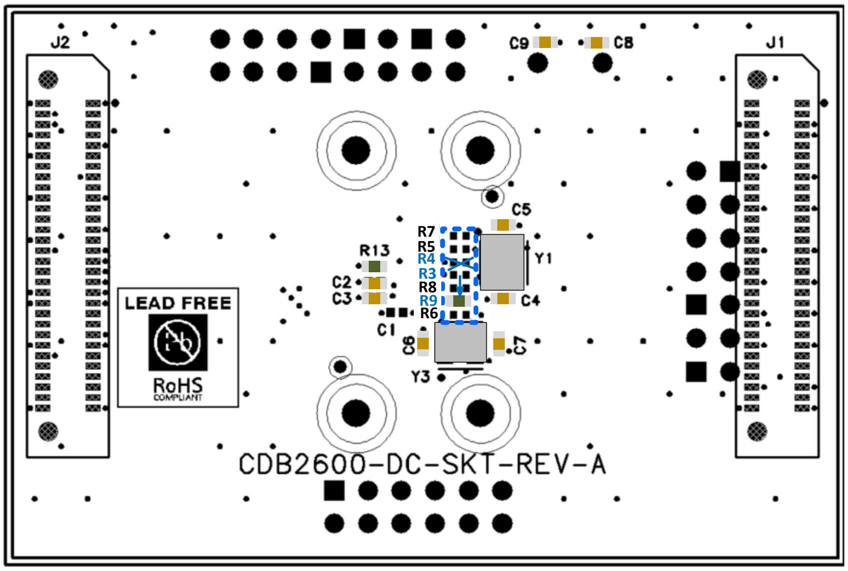


Figure 25 Daughter Card with External Reference Clock Option (R9)

- The REF_CLK can be provided from the internal LCO of the CS2600. This option is configured by moving jumpers P1 and P2 to positions 2 and 4.

  The configuration is illustrated in Figure 26.

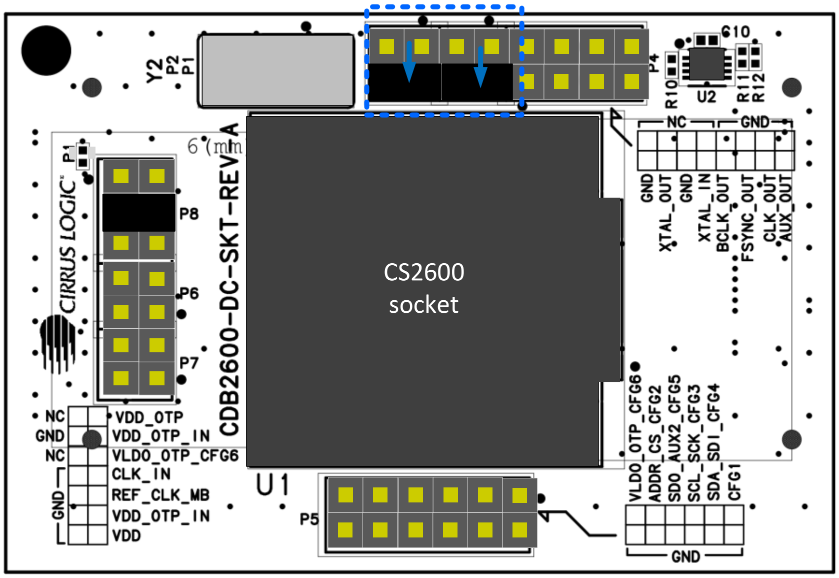


Figure 26 Daughter Card with Internal LCO as Reference Clock Option (R9)

- The REF_CLK can be provided from the 12 MHz crystal Y2 on the daughter card. The part number of the crystal is *ECS-120-18-4X-CKM* (through-hole crystal); further information can be found in the crystal datasheet.

  Crystal Y2 is provided in a socketed connector; other pin-compatible through-hole crystals can also be substituted.

  The daughter card configuration is R7, R8 populated (0 Ω); R3, R4, R5, R6, R9 non-populated.

  The configuration is illustrated in Figure 27.

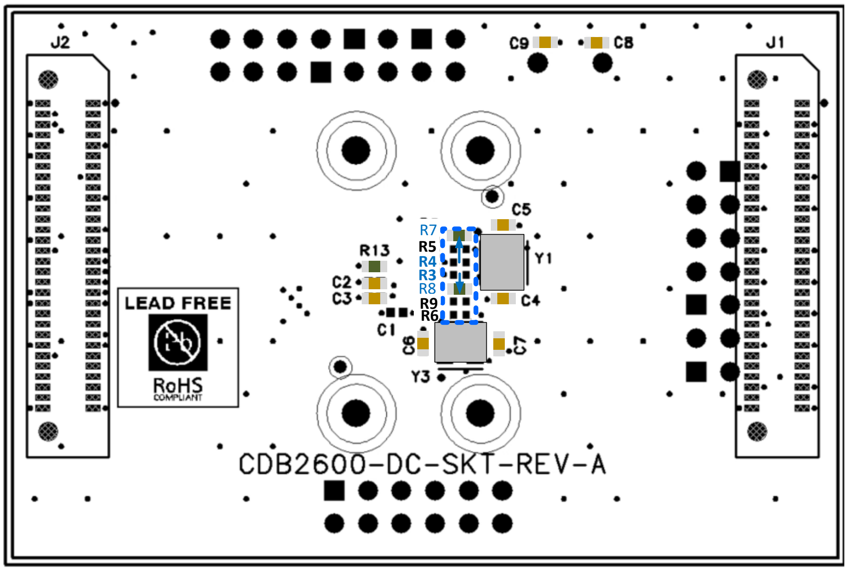


Figure 27 Daughter Card with Crystal Y2 Reference Clock Option (R7 and R8)

- The REF_CLK can be provided from the 12 MHz crystal Y3 on the daughter card. The part number of the crystal is *ECS-120-8-33Q-RES-TR* (SMD/SMT crystal); further information can be found in the crystal datasheet.

  The daughter card configuration is R5, R6 populated (0 Ω); R3, R4, R7, R8, R9 non-populated.

  The configuration is illustrated in Figure 28.

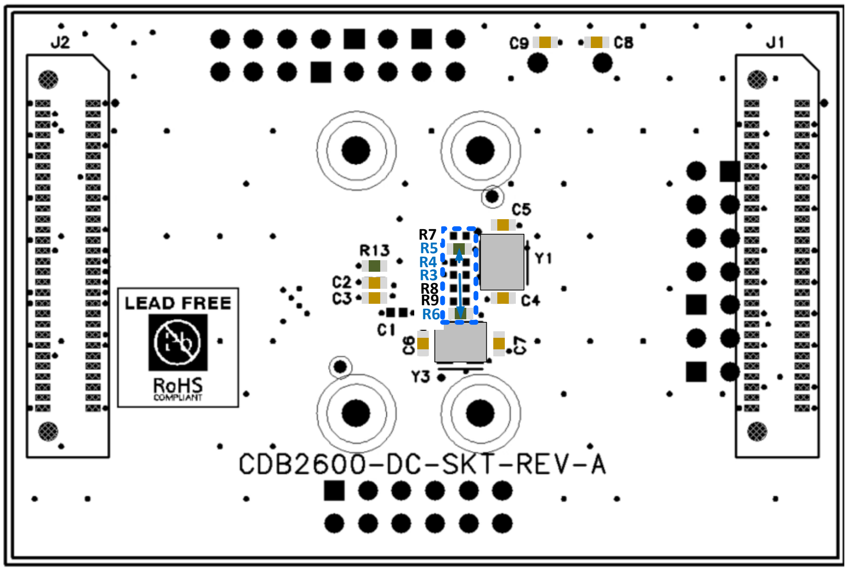


Figure 28 Daughter Card with Crystal Y3 as Reference Clock Option (R5 and R6)

5 Revision History

Revision History

Revision	Changes
R1 SEP 2024	• Initial version

Contacting Cirrus Logic Support
For all product questions and inquiries, contact a Cirrus Logic Sales Representative.
To find the one nearest you, go to www.cirrus.com.

IMPORTANT NOTICE

The products and services of Cirrus Logic International (UK) Limited; Cirrus Logic, Inc.; and other companies in the Cirrus Logic group (collectively either “Cirrus Logic” or “Cirrus”) are sold subject to Cirrus Logic’s terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, indemnification, and limitation of liability. Software is provided pursuant to applicable license terms. Cirrus Logic reserves the right to make changes to its products and specifications or to discontinue any product or service without notice. Customers should therefore obtain the latest version of relevant information from Cirrus Logic to verify that the information is current and complete. Testing and other quality control techniques are utilized to the extent Cirrus Logic deems necessary. Specific testing of all parameters of each device is not necessarily performed. In order to minimize risks associated with customer applications, the customer must use adequate design and operating safeguards to minimize inherent or procedural hazards. Cirrus Logic is not liable for applications assistance or customer product design. The customer is solely responsible for its overall product design, end-use applications, and system security, including the specific manner in which it uses Cirrus Logic components. Certain uses or product designs may require an intellectual property license from a third party. Features and operations described herein are for illustrative purposes only and do not constitute a suggestion or instruction to adopt a particular product design or a particular mode of operation for a Cirrus Logic component.

CIRRUS LOGIC PRODUCTS ARE NOT DESIGNED, TESTED, INTENDED OR WARRANTED FOR USE (1) WITH OR IN IMPLANTABLE PRODUCTS OR FDA/MHRA CLASS III (OR EQUIVALENT CLASSIFICATION) MEDICAL DEVICES, OR (2) IN ANY PRODUCTS, APPLICATIONS OR SYSTEMS, INCLUDING WITHOUT LIMITATION LIFE-CRITICAL MEDICAL EQUIPMENT OR SAFETY OR SECURITY EQUIPMENT, WHERE MALFUNCTION OF THE PRODUCT COULD CAUSE PERSONAL INJURY, DEATH, SEVERE PROPERTY DAMAGE OR SEVERE ENVIRONMENTAL HARM. INCLUSION OF CIRRUS LOGIC PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER’S RISK AND CIRRUS LOGIC DISCLAIMS AND MAKES NO WARRANTY, EXPRESS, STATUTORY OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE, WITH REGARD TO ANY CIRRUS LOGIC PRODUCT THAT IS USED IN SUCH A MANNER. IF THE CUSTOMER OR CUSTOMER’S CUSTOMER USES OR PERMITS THE USE OF CIRRUS LOGIC PRODUCTS IN SUCH A MANNER, CUSTOMER AGREES, BY SUCH USE, TO FULLY INDEMNIFY CIRRUS LOGIC, ITS OFFICERS, DIRECTORS, EMPLOYEES, DISTRIBUTORS AND OTHER AGENTS FROM ANY AND ALL LIABILITY, INCLUDING ATTORNEYS’ FEES AND COSTS, THAT MAY RESULT FROM OR ARISE IN CONNECTION WITH THESE USES.

This document is the property of Cirrus Logic, and you may not use this document in connection with any legal analysis concerning Cirrus Logic products described herein. No license to any technology or intellectual property right of Cirrus Logic or any third party is granted herein, including but not limited to any patent right, copyright, mask work right, or other intellectual property rights. Any provision or publication of any third party’s products or services does not constitute Cirrus Logic’s approval, license, warranty or endorsement thereof. Cirrus Logic gives consent for copies to be made of the information contained herein only for use within your organization with respect to Cirrus Logic integrated circuits or other products of Cirrus Logic, and only if the reproduction is without alteration and is accompanied by all associated copyright, proprietary and other notices and conditions (including this notice). This consent does not extend to other copying such as copying for general distribution, advertising or promotional purposes, or for creating any work for resale. This document and its information is provided “AS IS” without warranty of any kind (express or implied). All statutory warranties and conditions are excluded to the fullest extent possible. No responsibility is assumed by Cirrus Logic for the use of information herein, including use of this information as the basis for manufacture or sale of any items, or for infringement of patents or other rights of third parties. Cirrus Logic, Cirrus, the Cirrus Logic logo design, and SoundClear are among the trademarks of Cirrus Logic. Other brand and product names may be trademarks or service marks of their respective owners.