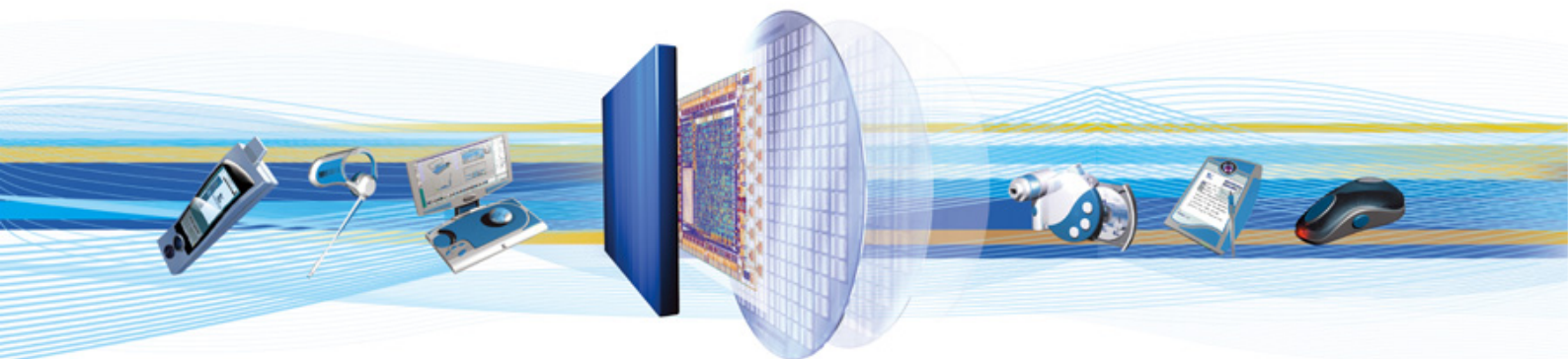




## CSR Synergy Bluetooth 18.2.0

# Resampling API Description

November 2011



### **Cambridge Silicon Radio Limited**

Churchill House  
Cambridge Business Park  
Cowley Road  
Cambridge CB4 0WZ  
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000

Fax: +44 (0)1223 692001

[www.csr.com](http://www.csr.com)



## Contents

<b>1</b>	<b>Introduction.....</b>	<b>3</b>
1.1	Introduction and Scope .....	3
1.2	Assumptions.....	3
<b>2</b>	<b>Description.....</b>	<b>4</b>
2.1	Introduction.....	4
2.2	Overview .....	4
<b>3</b>	<b>Interface Description.....</b>	<b>5</b>
3.1	CsrBtResamplingCfg.....	5
3.2	CsrBtResamplingDeInit.....	5
3.3	CsrBtResamplingStereo.....	5
3.4	CsrBtResamplingMono .....	6
3.5	CsrBtResamplingGetDataStereo .....	6
3.6	CsrBtResamplingGetDataMono.....	6
<b>4</b>	<b>Document References.....</b>	<b>7</b>

# 1 Introduction

## 1.1 Introduction and Scope

This document describes the function interface provided by the CSR Synergy Bluetooth Resampling implementation. The resampling implementation has the following functionality.

- Input and output frequency in the interval for 1Hz to -
- Input and output block size in the interval for 1 to  $2^{16}$
- Possible to convert from mono to stereo and from stereo to mono
- Possible to make multi-instances

## 1.2 Assumptions

The following assumptions and preconditions are made in the following:

- Audio samples are expected to be 16 bit wide

## 2 Description

### 2.1 Introduction

The resampling API is a low computational complexity algorithm specially designed to run on a 32 bit fixed-point general-purpose processor. To ensure low computational complexity the resampling is done with linear interpolation with additional low pass filters.

The resampling API supports a possible conversion between stereo and mono audio data.

### 2.2 Overview

The resampling API has the following functions:

- CsrBtResamplingDelInit
- CsrBtResamplingCfg
- CsrBtResamplingStereo
- CsrBtResamplingMono
- CsrBtResamplingGetDataStereo
- CsrBtResamplingGetDataMono

The next section describes the interface functions between the application and the resampling implementation.

## 3 Interface Description

### 3.1 CsrBtResamplingCfg

Prior to start of resampling, a call should be made to *ResamplingCfg* to initialize resampling and allocated memory to the instance data. If more than one resampling is needed resampling must be called again.

```
CsrBtResamplingHandler ResamplingCfg( ResamplingHandler inst,
                                       CsrInt32 fs0,
                                       CsrUInt8 bits0,
                                       CsrUInt8 chan0,
                                       CsrUInt32 blockSize0,
                                       CsrInt32 fs1,
                                       CsrUInt8 bits1,
                                       CsrUInt8 chan1,
                                       CsrUInt32 blockSize1,
                                       CsrUInt32 bufferSize);
```

#### Parameters

<i>ResamplingHandler</i>	Resampling handle
<i>CsrInt32 fs0</i>	Input sampling frequency
<i>CsrUInt8 bits0</i>	Input bits per sample. NOTE: Only implemented 16bit support
<i>CsrUInt8 chan0</i>	Input channel type: [0 = Mono and 1= Stereo]
<i>CsrUInt32 blockSize0</i>	Input block size [1 to 2 <sup>16</sup> ]
<i>CsrInt32 fs1</i>	Output sampling frequency
<i>CsrUInt8 bits1</i>	Output bits per sample NOTE: Only implemented 16bit support
<i>CsrUInt8 chan1</i>	Output channel type: [0 = Mono and 1= Stereo]
<i>CsrUInt32 blockSize1</i>	Output block size [1 to 2 <sup>16</sup> ]
<i>CsrUInt32 bufferSize</i>	Buffer size, is set to 0 the function calculate the needed buffersize

### 3.2 CsrBtResamplingDelInit

Resampling instance released.

```
void CsrBtResamplingDelInit( ResamplingHandler inst )
```

### 3.3 CsrBtResamplingStereo

Resampling of the audio data in dataChan0 and dataChan1, the re-sampled data is stored in a circular buffer. The block size can vary for call to call but it must not exceed the maximum block size set under configuration. The stereo data must be separated into two channels, dataChan0 and dataChan1.

```
void CsrBtResamplingStereo( ResamplingHandler inst,
                             CsrInt16 *dataChan0,
                             CsrInt16 *dataChan1,
                             CsrInt32 blockSize);
```

#### Parameters

<i>ResamplingHandler</i>	Instance data
<i>Int16_t</i>	Output data from channel 0
<i>Int16_t</i>	Output data from channel 1
<i>CsrInt32 blockSize</i>	Size of input data

### 3.4 CsrBtResamplingMono

Resampling of the audio data in dataChan0, the re-sampled data is stored in a circular buffer. The block size can vary for call to call but it must not exceed the maximum block size set under configuration.

```
void CsrBtResamplingMono(  ResamplingHandler inst,
                           CsrInt16 *dataChan0,
                           CsrInt32 blockSize);
```

#### Parameters

<i>ResamplingHandler</i>	Instance data
<i>Int16_t</i>	Output data from channel 0
<i>CsrInt32 blockSize</i>	Size of input data

### 3.5 CsrBtResamplingGetDataStereo

The resampling function puts the data into a buffer, to get the data out of the buffer ResamplingGetDataStereo must be used. Good practice is to call ResamplingGetDataStereo after each resampling call.

```
CsrUInt8 CsrBtResamplingGetDataStereo(  ResamplingHandler inst,
                                          CsrInt16 *dataChan0,
                                          CsrInt16 *dataChan1);
```

#### Parameters

<i>ResamplingHandler</i>	Instance data
<i>Int16_t</i>	Output data from channel 0
<i>Int16_t</i>	Output data from channel 1
Return value	Data access status [1 = success and 0 = Failed]

### 3.6 CsrBtResamplingGetDataMono

The resampling function puts the data into a buffer to get the data out of the buffer ResamplingGetDataStereo must be used. Good practice is to call ResamplingGetDataStereo after each resampling call.

```
CsrUInt8 CsrBtResamplingGetDataMono(  ResamplingHandler inst,
                                       CsrInt16 *dataChan0);
```

#### Parameters

<i>ResamplingHandler</i>	Instance data
<i>Int16_t</i>	Output data from channel 0
Return value	Data access status [1 = success and 0 = Failed]

## 4 Document References

Document	Reference
Specification of the Bluetooth System Version 1.1, 1.2 and 2.0	[BT]

## Terms and Definitions

AV	CSR Synergy Bluetooth Audio/Video profile component
BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
SNR	Signal-to-Noise Ratio
UniFi™	Group term for CSR's range of chips designed to meet IEEE 802.11 standards



## Document History

Revision	Date	History
1	26 SEP 11	Ready for release 18.2.0

## TradeMarks, Patents and Licences

Unless otherwise stated, words and logos marked with <sup>™</sup> or <sup>®</sup> are trademarks registered or owned by CSR plc or its affiliates. Bluetooth<sup>®</sup> and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR. Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

## Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

## Performance and Conformance

Refer to [www.csrsupport.com](http://www.csrsupport.com) for compliance and conformance to standards information.