

latex  
language=C++,inputencoding=utf8,basicstyle=,breaklines=true,breakatwhitespace=true,tabsize=4,numbers=left



kobanzame-sdk

1.30

Generated by Doxygen 1.6.2

Tue Jan 26 02:06:28 2010



# Contents



# Chapter 1

## KOBANZAME SDK

### Software Developers Kit for KOBANZAME Blackfin Evaluation Board

This software is free and is developed in SourceForge.jp KOBANZAME SDK Project. Please visit project page for getting latest software and documents(Japanese).

<http://sourceforge.jp/projects/kobanzame-sdk/>

**KOBANZAME SDK** is a Software Developers Kit(SDK) for KOBANZAME DSP Evaluation Board which embedded Analog Devices Blackfin Processor BF-533.

Target users of this software are those who

- Try to learn DSP Processor
- Try DSP evaluation quickly
- Implement your Audio DSP algorithm quickly
- etc ...

This SDK has following features.

- uITRON4 RTOS is ready to use. Users can implement multi-task system easily.
- Filesystem(FAT32/16/12/VFAT) for micro-SD-Card is ready to use.
- Serial Command line is ready to use. Users can add original commands easily.
- Device Drivers APIs are ready to use. Users can access following hardwares easily.
  - 48kHz 24bit 4in/ 4out audio
  - MEMS
  - Switches
  - LEDs
- The SDK equips almost C-Language standard functions defined in stdio.h. Thier standard I/O is accessed via Serial or micro-SD's filesystem, not debugger console. The functions are overwritten standard ones by including "kobanzame.h".
- Q5.26 Audio DSP Library is ready to use. The library uses Blackfin Hardware DSP Units, and users can make powerful DSP software easily even without knowledge of DSP Hardware Units.

The SDK support following tool.

- Analog Devices VDSP++5.0 Update7

KOBANZAME is a DSP evaluation board distributed by J-Person Co., Ltd. Main parts of the board are

- Analog Devices Blackfin Processor BF-533 500MHz
- 32Mbyte SD-RAM
- 2MByte Serial flash
- Analog Devices Audio CODEC AD1836
- Freescale MEMS MMA7455L
- Two LEDs
- Two Switches

Get more information about KOBANZAME, please visit j-person's home page(English).

<http://kobanzame.j-person.com/bf533/index-en.html>

This software is consist of

- Toppers JSP(RTOS) by TOPPERS Project
- Toppers JSP for Blackfin by suikan
- FAT Filesystem Fatfs by ChaN
- Device Drivers for KOBANZAME
- Command line

**LICENSE:** The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

The documents are automatically generated by DOXYGEN from the SDK source codes

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved



# Chapter 2

## Module Index

### 2.1 Modules

Here is a list of all modules:

API_DeviceDriver . . . . .	??
API_CommandLine . . . . .	??
API_stdio . . . . .	??
API_AudioDSP . . . . .	??
UserCongifuration . . . . .	??



# Chapter 3

## Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">file_t</a> . . . . .	??
<a href="#">KzCmdRegist_t</a> . . . . .	??
<a href="#">SpiDeviceConfigurator_t</a> . . . . .	??



# Chapter 4

## File Index

### 4.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">cmdline.c</a> (Command line ) . . . . .	??
<a href="#">def_mma7455l.h</a> (Freescale MMA7455L MEMS Define ) . . . . .	??
<a href="#">fs_cmd.c</a> (Filesystem relative command line ) . . . . .	??
<a href="#">fs_support.c</a> (Filesystem support functions ) . . . . .	??
<a href="#">kobanzame.h</a> (KOBANZAME SDK Standard Include file ) . . . . .	??
<a href="#">kz_malloc.c</a> (Re-entrant malloc support ) . . . . .	??
<a href="#">kzconfig.h</a> (KOBANZAME SDK USER configuration file ) . . . . .	??
<a href="#">kzdev.c</a> (KOBANZAME SDK Device Drivers ) . . . . .	??
<a href="#">kzdev.h</a> (KOBANZAME SDK Device Driver ) . . . . .	??
<a href="#">kzdev_audio.c</a> (KOBANZAME SDK Audio Driver ) . . . . .	??
<a href="#">kzdev_codec.c</a> (KOBANZAME SDK Audio Codec Driver ) . . . . .	??
<a href="#">kzdev_mems.c</a> (KOBANZAME SDK MEMS Device Driver ) . . . . .	??
<a href="#">kzdev_spi.c</a> (KOBANZAME SDK SPI Device Driver ) . . . . .	??
<a href="#">kzdsp.h</a> (DSP Libray ) . . . . .	??
<a href="#">kzprivate.h</a> (SDK Private Global Function define ) . . . . .	??
<a href="#">kzstdio.c</a> (KOBANZAME SDK stdio function ) . . . . .	??
<a href="#">kzstdio.h</a> (KOBANZAME SDK stdio function ) . . . . .	??
<a href="#">kzversion.h</a> (KOBANZAME SDK Version File ) . . . . .	??
<a href="#">serial_io.c</a> (KOBANZAME SDK Serial I/O function ) . . . . .	??
<a href="#">target_def.h</a> (KOBANZAME SDK Target selector ) . . . . .	??



# Chapter 5

## Module Documentation

### 5.1 API\_DeviceDriver

#### Functions

- [KZSTATUS\\_t KzSwStart \(KzSwCbk\\_t cbk\)](#)  
*Swith start.*
- [KZSTATUS\\_t KzSwStop \(void\)](#)  
*Swith stop.*
- [KZSTATUS\\_t KzLedOn \(KzLED\\_t nLedNo\)](#)  
*Set LED ON.*
- [KZSTATUS\\_t KzLedOff \(KzLED\\_t nLedNo\)](#)  
*Set LED OFF.*
- [KZSTATUS\\_t KzLedBlink \(KzLED\\_t nLedNo, int nOnTime, int nOffTime\)](#)  
*Set LED blink.*
- [KZSTATUS\\_t KzMemsStart \(KzMemsCbk\\_t cbk\)](#)  
*MEMS start.*
- [KZSTATUS\\_t KzMemsStop \(void\)](#)  
*MEMS stop.*
- [KZSTATUS\\_t KzAudioStart \(KzAudioCbk\\_t cbk, long lSampleRate, int nBlocks, int nChannels\)](#)  
*Audio start.*
- [KZSTATUS\\_t KzAudioStop \(void\)](#)  
*Audio stop.*

## 5.1.1 Function Documentation

### 5.1.1.1 KZSTATUS\_t KzAudioStart (KzAudioCbk\_t *cbk*, long *lSampleRate*, int *nBlocks*, int *nChannels*)

Audio start.

#### Parameters:

*cbk* User callback function. This function is called in Audio Task(highest priority) every nBlocks.

*lSampleRate* Audio sample rate. now you can set only 48000.

*nBlocks* Audio blocks that callback function has. This must be a multiple of 8(SAMPLES\_PER\_INTR defined in [kzdev\\_audio.c](#)).

*nChannels* numoer of the Audio Channels (1-4)

#### Returns:

KZ\_OK Success

KZ\_ERR Fail to start Audio

### 5.1.1.2 KZSTATUS\_t KzAudioStop (void)

Audio stop.

#### Returns:

KZ\_OK Success (constant)

### 5.1.1.3 KZSTATUS\_t KzLedBlink (KzLED\_t *nLedNo*, int *nOnTime*, int *nOffTime*)

Set LED blink.

#### Parameters:

*nLedNo* KOBANZAME hardware LED number

*nOnTime* Interval of LED Lit time (ms)

*nOffTime* Interval of LED Off time (ms)

#### Returns:

KZ\_OK Success

KZ\_ERR Undefined led number

### 5.1.1.4 KZSTATUS\_t KzLedOff (KzLED\_t *nLedNo*)

Set LED OFF.

#### Parameters:

*nLedNo* KOBANZAME hardware LED number

#### Returns:

KZ\_OK Success

KZ\_ERR Undefined led number



**5.1.1.5 KZSTATUS\_t KzLedOn (KzLED\_t nLedNo)**

Set LED ON.

**Parameters:**

*nLedNo* KOBANZAME hardware LED number

**Returns:**

KZ\_OK Success  
KZ\_ERR Undefined led number

**5.1.1.6 KZSTATUS\_t KzMemsStart (KzMemsCbK\_t cbk)**

MEMS start.

**Parameters:**

*cbk* User callback function. This function is called in DeviceDriver(middle priority) every MEMS Status changes.

**Returns:**

KZ\_OK Success  
KZ\_ERR Fail to start MEMS

**5.1.1.7 KZSTATUS\_t KzMemsStop (void)**

MEMS stop.

**Returns:**

KZ\_OK Success  
KZ\_ERR Fail to stop

**5.1.1.8 KZSTATUS\_t KzSwStart (KzSwCbK\_t cbk)**

Swith start.

**Parameters:**

*cbk* User callback function. This function is called in DeviceDriver(middle priority) every switches Status changes.

**Returns:**

KZ\_OK Success  
KZ\_ERR Fail to start switch

**5.1.1.9 KZSTATUS\_t KzSwStop (void)**

Switth stop.

**Returns:**

KZ\_OK Success  
KZ\_ERR Fail to stop switch

## 5.2 API\_CommandLine

### Functions

- [KZSTATUS\\_t KzCmdlineStart](#) (void)  
*Command Line Start.*
- [KZSTATUS\\_t KzCmdlineExit](#) (void)  
*Command Line Exit.*
- [KZSTATUS\\_t KzCmdlineAdd](#) (const [KzCmdRegist\\_t](#) \*exe)  
*Add a Command.*
- [KZSTATUS\\_t KzCmdlineAddMany](#) (const [KzCmdRegist\\_t](#) \*exes)  
*Add Many Commands.*
- [KZSTATUS\\_t KzFilesystemStart](#) (void)  
*Filesystem start.*
- [KZSTATUS\\_t KzAddCmdFilesystem](#) (void)  
*Add Filesystem commands.*
- [KZSTATUS\\_t KzAddCmdDeviceDriver](#) (void)  
*Add DeviceDriver commands.*

### 5.2.1 Function Documentation

#### 5.2.1.1 KZSTATUS\_t KzAddCmdDeviceDriver (void)

Add DeviceDriver commands.

#### Returns:

KZ\_OK Success  
Lack of memory space for the command. See [KzCmdlineAddMany](#)

#### Note:

After this function call. User can use following commands.

- swled
- wink
- mems
- echo

### 5.2.1.2 KZSTATUS\_t KzAddCmdFilesystem (void)

Add Filesystem commands.

**Returns:**

KZ\_OK Success

Lack of memory space for the command. See KzCmdlineAddMany

**Note:**

After this function call. User can use following commands.

- ls
- pwd
- cd
- mkdir
- rmdir
- cp
- rm
- mv
- dump
- more

### 5.2.1.3 KZSTATUS\_t KzCmdlineAdd (const KzCmdRegist\_t \* exe)

Add a Command.

**Parameters:**

*exe* add command

**Returns:**

KZ\_OK Success

KZ\_ERR Lack of memory space for the command

**Note:**

if KZ\_ERR is occurred, you can revise KZCMDLINE\_MAX\_COMMANDS in [kzconfig.h](#) to increase the number of the command line.

### 5.2.1.4 KZSTATUS\_t KzCmdlineAddMany (const KzCmdRegist\_t \* exe)

Add Many Commands.

**Parameters:**

*exe* add command, must be terminated by ZERO

**Returns:**

KZ\_OK Success

KZ\_ERR Lack of memory space for the command

**Note:**

if KZ\_ERR is occurred, you can revise KZCMDLINE\_MAX\_COMMANDS in [kzconfig.h](#) to increase the number of the command line.

**5.2.1.5 KZSTATUS\_t KzCmdlineExit (void)**

Command Line Exit.

**Returns:**

KZ\_OK Success (constant)

**5.2.1.6 KZSTATUS\_t KzCmdlineStart (void)**

Command Line Start.

**Returns:**

KZ\_OK Success (constant)

**Note:**

The function does not return until [KzCmdlineExit\(\)](#) is called

**5.2.1.7 KZSTATUS\_t KzFilesystemStart (void)**

Filesystem start.

**Returns:**

KZ\_OK Success

KZ\_ERR Filesystem cannot start. ( mainly SD-Card is not inserted )

**Warning:**

Before this function call. User must insert SD-Card, otherwise filesystem won't start and this function returns KZ\_ERR.

## 5.3 API\_stdio

### Functions

- int [Kz\\_fgetc](#) (FILE \*fp)  
*Standard C Compatible function: fgetc.*
- int [Kz\\_fputc](#) (int c, FILE \*fp)  
*Standard C Compatible function: fputc.*
- FILE \* [Kz\\_fopen](#) (const char \*filename, const char \*mode)  
*Standard C Compatible function: fopen.*
- int [Kz\\_fclose](#) (FILE \*fp)  
*Standard C Compatible function: fclose.*
- int [Kz\\_fseek](#) (FILE \*fp, long offset, int whence)  
*Standard C Compatible function: fseek.*
- size\_t [Kz\\_fread](#) (void \*ptr, size\_t size, size\_t nmemb, FILE \*stream)  
*Standard C Compatible function: fread.*
- size\_t [Kz\\_fwrite](#) (const void \*ptr, size\_t size, size\_t nmemb, FILE \*stream)  
*Standard C Compatible function: fwrite.*
- char \* [Kz\\_fgets](#) (char \*s, int n, FILE \*fp)  
*Standard C Compatible function: fgets.*
- int [Kz\\_fputs](#) (const char \*s, FILE \*fp)  
*Standard C Compatible function: fputs.*
- int [Kz\\_fscanf](#) (FILE \*fp, const char \*format,...)  
*Standard C Compatible function: fscanf.*
- int [Kz\\_fprintf](#) (FILE \*fp, const char \*format,...)  
*Standard C Compatible function: fprintf.*
- int [Kz\\_printf](#) (const char \*format,...)  
*Standard C Compatible function: printf.*
- int [Kz\\_scanf](#) (const char \*format,...)  
*Standard C Compatible function: scanf.*
- char \* [Kz\\_gets](#) (char \*s)  
*Standard C Compatible function: gets.*
- int [Kz\\_puts](#) (const char \*s)  
*Standard C Compatible function: puts.*
- int [Kz\\_vprintf](#) (const char \*format, va\_list arg)

*Standard C Compatible function: vprintf.*

- int [Kz\\_vfprintf](#) (FILE \*fp, const char \*format, va\_list arg)  
*Standard C Compatible function: vfprintf.*
- int [Kz\\_getchar](#) (void)  
*Standard C Compatible function: getchar.*
- int [Kz\\_putchar](#) (int c)  
*Standard C Compatible function: putchar.*
- int [Kz\\_getc](#) (FILE \*fp)  
*Standard C Compatible function:getc.*
- int [Kz\\_putc](#) (int c, FILE \*fp)  
*Standard C Compatible function:putc.*

### 5.3.1 Function Documentation

#### 5.3.1.1 int [Kz\\_fclose](#) (FILE \*fp)

Standard C Compatible function: fclose.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

#### 5.3.1.2 int [Kz\\_fgetc](#) (FILE \*fp)

Standard C Compatible function: fgetc.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

#### 5.3.1.3 char \* [Kz\\_fgets](#) (char \*s, int n, FILE \*fp)

Standard C Compatible function: fgets.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.4 FILE \* Kz\_fopen (const char \* *filename*, const char \* *mode*)**

Standard C Compatible function: fopen.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.5 int Kz\_fprintf (FILE \* *fp*, const char \* *format*, ...)**

Standard C Compatible function: fprintf.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.6 int Kz\_fputc (int *c*, FILE \* *fp*)**

Standard C Compatible function: fputc.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.7 int Kz\_fputs (const char \* *s*, FILE \* *fp*)**

Standard C Compatible function: fputs.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)



**5.3.1.8** `size_t Kz_fread (void * ptr, size_t size, size_t nmemb, FILE * stream)`

Standard C Compatible function: fread.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.9** `int Kz_fscanf (FILE * fp, const char * format, ...)`

Standard C Compatible function: fscanf.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.10** `int Kz_fseek (FILE * fp, long offset, int whence)`

Standard C Compatible function: fseek.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.11** `size_t Kz_fwrite (const void * ptr, size_t size, size_t nmemb, FILE * stream)`

Standard C Compatible function: fwrite.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.12** `iiint Kz_getc (FILE * fp)`

Standard C Compatible function: `getc`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.13** `int Kz_getchar (void)`

Standard C Compatible function: `getchar`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.14** `char * Kz_gets (char * s)`

Standard C Compatible function: `gets`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.15** `int Kz_printf (const char * format, ...)`

Standard C Compatible function: `printf`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.16 int Kz\_putc (int *c*, FILE \**fp*)**

Standard C Compatible function: `putc`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.17 int Kz\_putchar (int *c*)**

Standard C Compatible function: `putchar`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.18 int Kz\_puts (const char \**s*)**

Standard C Compatible function: `puts`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.19 int Kz\_scanf (const char \**format*, ...)**

Standard C Compatible function: `scanf`.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.20 int Kz\_vfprintf (FILE \**fp*, const char \**format*, va\_list *arg*)**

Standard C Compatible function: fprintf.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

**5.3.1.21 int Kz\_vprintf (const char \**format*, va\_list *arg*)**

Standard C Compatible function: printf.

**See also:**

General Standard C documents

**Note:**

Renamed to standard symbol name in [kzstdio.h](#)

## 5.4 API\_AudioDSP

### Functions

- int [KzDspAdd](#) (int a, int b)  
*a + b with saturation*
- int [KzDspSub](#) (int a, int b)  
*a - b with saturation*
- int [KzDspMpy](#) (int a, int b)  
*a \* b with saturation*
- int [KzDspMac](#) (int \*a, int \*b, int n)  
*Multiply and Accumrate.*
- int [KzDspSat](#) (int a)  
*Saturation.*
- int [KzDspAbs](#) (int a)  
*Absolute.*
- int [KzDspShl](#) (int a, int s)  
*Shift left with saturation.*
- int [KzDspShr](#) (int a, int s)  
*Shift right with saturation.*
- int [KzDspNeg](#) (int a)  
*Nagate.*
- float [KzDspTof](#) (int a)  
*To float.*
- int [KzDspToi](#) (float a)  
*To int.*

### 5.4.1 Function Documentation

#### 5.4.1.1 int [KzDspAbs](#) (int a) [`inline`]

Absolute.

#### Parameters:

*a* parameter1

#### Returns:

if ( a<0.0) ans=-a; else ans = a;

**5.4.1.2 int KzDspAdd (int *a*, int *b*) [inline]**

$a + b$  with saturation

**Parameters:**

*a* parameter1  
*b* parameter2

**Returns:**

$a + b$  with saturation

**5.4.1.3 int KzDspMac (int \* *a*, int \* *b*, int *n*) [inline]**

Multiply and Accumrate.

**Parameters:**

*a* parameter1  
*b* parameter2  
*n* loops

**Returns:**

$ans += *a++ * *b++;$  loop *n*

**5.4.1.4 int KzDspMpy (int *a*, int *b*) [inline]**

$a * b$  with saturation

**Parameters:**

*a* parameter1  
*b* parameter2

**Returns:**

$a * b$  with saturation

**See also:**

<http://blackfin.s36.coreserver.jp/index.php?id=89>

**5.4.1.5 int KzDspNeg (int *a*) [inline]**

Nagate.

**Parameters:**

*a* parameter1

**Returns:**

-*a*

**5.4.1.6 int KzDspSat (int *a*) [inline]**

Saturation.

**Parameters:**

*a* parameter1

**Returns:**

saturation limit is [-1.0,1.0]

**Note:**

if ( $a > 1.0$ ) ans = 1.0; if ( $a < -1.0$ ) ans = -1.0; else ans = *a*;

**5.4.1.7 int KzDspShl (int *a*, int *s*) [inline]**

Shift left with saturation.

**Parameters:**

*a* parameter1

*s* shift value ( minus value is acceptable )

**Returns:**

$a \ll s$

**5.4.1.8 int KzDspShr (int *a*, int *s*) [inline]**

Shift right with saturation.

**Parameters:**

*a* parameter1

*s* shift value ( minus value is acceptable )

**Returns:**

$a \gg s$

**5.4.1.9 int KzDspSub (int *a*, int *b*) [inline]**

$a - b$  with saturation

**Parameters:**

*a* parameter1

*b* parameter2

**Returns:**

$a - b$  with saturation

**5.4.1.10 float KzDspTof (int *a*) [inline]**

To float.

**Parameters:**

*a* parameter1

**Returns:**

(float)*a*

**5.4.1.11 int KzDspToi (float *a*) [inline]**

To int.

**Parameters:**

*a* parameter1

**Returns:**

(int)*a*



## 5.5 UserCongifuration

### Defines

- #define [KZFILE\\_MAX\\_OPEN](#) (10)  
*Max files that filesystem can open.*
- #define [KZFILE\\_MAX\\_PATH\\_LEN](#) (512)  
*Max file path length.*
- #define [KZCMDLINE\\_MAX\\_INPUT\\_CHAR](#) (128)  
*Max Command Line charactor length.*
- #define [KZCMDLINE\\_MAX\\_COMMANDS](#) (32)  
*Max allocate number of the commands.*
- #define [KZCMDLINE\\_MAX\\_ARGS](#) (16)  
*Max arguments that command line execute function can receive.*
- #define [KZAUDIO\\_MAX\\_AUDIO\\_NBLOCKS](#) (32)  
*Max Audio buffer blocks that callback can receive.*
- #define [KZUSING\\_SD\\_BENCH\\_TEST](#) (1)  
*(1) add SD-Card bench mark test command in filesystem command*



# Chapter 6

## Data Structure Documentation

### 6.1 file\_t Struct Reference

#### Data Fields

- FILE [mFile](#)
- BOOL [mbUsed](#)

#### 6.1.1 Detailed Description

file object type

#### 6.1.2 Field Documentation

##### 6.1.2.1 BOOL mbUsed

file object is used or not

##### 6.1.2.2 FILE mFile

file object

The documentation for this struct was generated from the following file:

- [kzstdio.c](#)

## 6.2 KzCmdRegist\_t Struct Reference

```
#include <kobanzame.h>
```

### Data Fields

- [KzFnCmd\\_t mfnExe](#)
- const char \* [msCmd](#)
- const char \* [msHelp](#)

### 6.2.1 Detailed Description

command line registration type

### 6.2.2 Field Documentation

#### 6.2.2.1 KzFnCmd\_t mfnExe

function pointer for the command line

#### 6.2.2.2 const char\* msCmd

command strings for the command line

#### 6.2.2.3 const char\* msHelp

help strings for the command line

The documentation for this struct was generated from the following file:

- [kobanzame.h](#)

## 6.3 SpiDeviceConfigurator\_t Struct Reference

### Data Fields

- int [mnCS](#)
- UW [mdwBPS](#)
- KZDEV\_SPI\_BITLEN\_t [mnDeviceBit](#)
- BOOL [mbCPOL](#)
- BOOL [mbCPHA](#)

### 6.3.1 Field Documentation

#### 6.3.1.1 BOOL mbCPHA

SPI CPHA set or not

#### 6.3.1.2 BOOL mbCPOL

SPI CPOL set or not

#### 6.3.1.3 UW mdwBPS

bit per second

#### 6.3.1.4 int mnCS

chip select number

#### 6.3.1.5 KZDEV\_SPI\_BITLEN\_t mnDeviceBit

8bit or 16bit

The documentation for this struct was generated from the following file:

- [kzdev\\_spi.c](#)



# Chapter 7

## File Documentation

### 7.1 cmdline.c File Reference

```
command line #include "kobanzame.h"
```

```
#include <string.h>
```

```
#include "serial.h"
```

Include dependency graph for cmdline.c:

#### Functions

- [KZSTATUS\\_t KzCmdlineStart](#) (void)  
*Command Line Start.*
- [KZSTATUS\\_t KzCmdlineExit](#) (void)  
*Command Line Exit.*
- [KZSTATUS\\_t KzCmdlineAdd](#) (const [KzCmdRegist\\_t](#) \*exe)  
*Add a Command.*
- [KZSTATUS\\_t KzCmdlineAddMany](#) (const [KzCmdRegist\\_t](#) \*exes)  
*Add Many Commands.*

#### 7.1.1 Detailed Description

command line KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

## 7.2 def\_mma7455l.h File Reference

Freescale MMA7455L MEMS Define. This graph shows which files directly or indirectly include this file:

### Defines

- #define **REG\_MMA7455L\_XOUTL** (0x00)
- #define **REG\_MMA7455L\_XOUTH** (0x01)
- #define **REG\_MMA7455L\_YOUTL** (0x02)
- #define **REG\_MMA7455L\_YOUTH** (0x03)
- #define **REG\_MMA7455L\_ZOUTL** (0x04)
- #define **REG\_MMA7455L\_ZOUTH** (0x05)
- #define **REG\_MMA7455L\_XOUT8** (0x06)
- #define **REG\_MMA7455L\_YOUT8** (0x07)
- #define **REG\_MMA7455L\_ZOUT8** (0x08)
- #define **REG\_MMA7455L\_STATUS** (0x09)
- #define **REG\_MMA7455L\_DETSRC** (0x0A)
- #define **REG\_MMA7455L\_TOUT** (0x0B)
- #define **REG\_MMA7455L\_RESERVED0** (0x0C)
- #define **REG\_MMA7455L\_I2CAD** (0x0D)
- #define **REG\_MMA7455L\_USRINF** (0x0E)
- #define **REG\_MMA7455L\_WHOAMI** (0x0F)
- #define **REG\_MMA7455L\_XOFFL** (0x10)
- #define **REG\_MMA7455L\_XOFFH** (0x11)
- #define **REG\_MMA7455L\_YOFFL** (0x12)
- #define **REG\_MMA7455L\_YOFFH** (0x13)
- #define **REG\_MMA7455L\_ZOFFL** (0x14)
- #define **REG\_MMA7455L\_ZOFFH** (0x15)
- #define **REG\_MMA7455L\_MCTL** (0x16)
- #define **REG\_MMA7455L\_INTRST** (0x17)
- #define **REG\_MMA7455L\_CTL1** (0x18)
- #define **REG\_MMA7455L\_CTL2** (0x19)
- #define **REG\_MMA7455L\_LDTH** (0x1A)
- #define **REG\_MMA7455L\_PDTH** (0x1B)
- #define **REG\_MMA7455L\_PW** (0x1C)
- #define **REG\_MMA7455L\_LT** (0x1D)
- #define **REG\_MMA7455L\_TW** (0x1E)
- #define **REG\_MMA7455L\_RESERVED1** (0x1F)
- #define **BITDEF\_MMA7455L\_DRDY** (1<<0)
- #define **BITDEF\_MMA7455L\_DOVR** (1<<1)
- #define **BITDEF\_MMA7455L\_PERR** (1<<2)
- #define **BITDEF\_MMA7455L\_INT1** (1<<0)
- #define **BITDEF\_MMA7455L\_INT2** (1<<1)
- #define **BITDEF\_MMA7455L\_PDZ** (1<<2)
- #define **BITDEF\_MMA7455L\_PDY** (1<<3)
- #define **BITDEF\_MMA7455L\_PDX** (1<<4)
- #define **BITDEF\_MMA7455L\_LDZ** (1<<5)
- #define **BITDEF\_MMA7455L\_LDY** (1<<6)



- #define **BITDEF\_MMA7455L\_LDX** (1<<7)
- #define **BITDEF\_MMA7455L\_I2CDIS** (1<<7)
- #define **BITMASK\_MMA7455L\_MODE** (0x3<<0)
- #define **BITATTR\_MMA7455L\_STANBY\_MODE** (0<<0)
- #define **BITATTR\_MMA7455L\_MEASURE\_MODE** (1<<0)
- #define **BITATTR\_MMA7455L\_LEVEL\_MODE** (2<<0)
- #define **BITATTR\_MMA7455L\_PULSE\_MODE** (3<<0)
- #define **BITMASK\_MMA7455L\_GLVL** (0x3<<2)
- #define **BITATTR\_MMA7455L\_8G** (0<<2)
- #define **BITATTR\_MMA7455L\_4G** (1<<2)
- #define **BITATTR\_MMA7455L\_2G** (2<<2)
- #define **BITDEF\_MMA7455L\_STON** (1<<4)
- #define **BITDEF\_MMA7455L\_SPI3W** (1<<5)
- #define **BITDEF\_MMA7455L\_DRPD** (1<<6)
- #define **BITDEF\_MMA7455L\_CLR\_INT1** (1<<0)
- #define **BITDEF\_MMA7455L\_CLR\_INT2** (1<<1)
- #define **BITDEF\_MMA7455L\_INTPIN** (1<<0)
- #define **BITDEF\_MMA7455L\_INTREG0** (1<<1)
- #define **BITDEF\_MMA7455L\_INTREG1** (1<<2)
- #define **BITDEF\_MMA7455L\_XDA** (1<<3)
- #define **BITDEF\_MMA7455L\_YDA** (1<<4)
- #define **BITDEF\_MMA7455L\_ZDA** (1<<5)
- #define **BITDEF\_MMA7455L\_THOPT** (1<<6)
- #define **BITDEF\_MMA7455L\_DFBW** (1<<7)
- #define **BITDEF\_MMA7455L\_LDPL** (1<<0)
- #define **BITDEF\_MMA7455L\_PDPL** (1<<1)
- #define **BITDEF\_MMA7455L\_DRVO** (1<<2)

### 7.2.1 Detailed Description

Freescale MMA7455L MEMS Define. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### Note:

MMA7455L(freescale MEMS) register define

## 7.3 fs\_cmd.c File Reference

filesystem relative command line `#include "kobanzame.h"`

`#include <string.h>`

Include dependency graph for fs\_cmd.c:

### Defines

- `#define MORE_MAX_LIST_LINES (32)`  
*USER DEFINE: Max lines at more command.*
- `#define TEST_FILE_SIZE (1024*1024*10)`
- `#define TEST_FILE_NAME "test_tmp"`
- `#define SD_TEST_VERSION 1`
- `#define TEST_PTN_4BYTE 0x55565758L`

### Functions

- `const char * KzGetCurPath (void)`
- `void KzSetPath (char *dest, const char *arg)`  
*SDK Private : Set file path.*
- `KZSTATUS_t KzFilesystemStart (void)`  
*Filesystem start.*
- `KZSTATUS_t KzAddCmdFilesystem (void)`  
*Add Filesystem commands.*

#### 7.3.1 Detailed Description

filesystem relative command line KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### 7.3.2 Function Documentation

##### 7.3.2.1 void KzSetPath (char \* dest, const char \* arg)

SDK Private : Set file path.

**Parameters:**

*dest* destination buffer

*arg* path argument

## 7.4 fs\_support.c File Reference

filesystem support functions `#include "kobanzame.h"`

Include dependency graph for fs\_support.c:

### Functions

- void **KzAttIniFatfs** (ID semid)
- DWORD **get\_fattime** (void)
- BOOL **ff\_cre\_syncobj** (BYTE a, \_SYNC\_t \*s)
- BOOL **ff\_del\_syncobj** (\_SYNC\_t s)
- BOOL **ff\_req\_grant** (\_SYNC\_t s)
- void **ff\_rel\_grant** (\_SYNC\_t s)

### 7.4.1 Detailed Description

filesystem support functions KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### Note:

filesystem re-entrant support functions

## 7.5 kobanzame.h File Reference

```
KOBANZAME SDK Standard Include file. #include "target_def.h"
#include "kzconfig.h"
#include "kernel.h"
#include <stdio.h>
#include "kzstdio.h"
#include "kzdev.h"
#include "kzdsp.h"
#include "kzprivate.h"
```

Include dependency graph for kobanzame.h:

This graph shows which files directly or indirectly include this file:

### Data Structures

- struct [KzCmdRegist\\_t](#)

### Defines

- #define [numof\(x\)](#) sizeof(x)/sizeof(x[0])
- #define [KZ\\_AUDIOCLUSTAR](#) (4)  
*Numbers of the Audio Clustar, Audio Clustar Order is L0/R0/L1/R1.*
- #define [L1DATA\\_S](#) section("L1\_data\_s")  
*Memory section directive: L1 SCRATCH.*
- #define [L1DATA\\_A](#) section("L1\_data\_a")  
*Memory section directive: L1 DATA A.*
- #define [L1DATA\\_B](#) section("L1\_data\_b")  
*Memory section directive: L1 DATA B.*
- #define [L1CODE](#) section("L1\_code")  
*Memory section directive: L1 CODE.*
- #define [L3DATA](#) section("L3\_data")  
*Memory section directive: SDRAM DATA.*
- #define [L3CODE](#) section("L3\_code")  
*Memory section directive: SDRAM CODE.*

## Typedefs

- typedef int(\* [KzFnCmd\\_t](#))(int argc, char \*argv[ ])
   
*command line function's type*
- typedef void(\* [KzSwCbk\\_t](#))([KzSW\\_t](#) nSw, BOOL bPushed)
   
*callback type for physical switches change status*
- typedef void(\* [KzMemsCbk\\_t](#))(int x, int y, int z)
   
*callback function type for MEMS changes*
- typedef void(\* [KzAudioCbk\\_t](#))(int \*in, int \*out, int nBlocks, int nChannels)
   
*callback function type for audio*

## Enumerations

- enum [KzSW\\_t](#) { [eKzSW0](#), [eKzSW1](#), [numof\\_KzSW](#) }
- enum [KzLED\\_t](#) { [eKzLED0](#), [eKzLED1](#), [numof\\_KzLED](#) }
- enum [KZSTATUS\\_t](#) { [KZ\\_OK](#) = 0, [KZ\\_ERR](#) = -1 }

## Functions

- [KZSTATUS\\_t](#) [KzCmdlineStart](#) (void)
   
*Command Line Start.*
- [KZSTATUS\\_t](#) [KzCmdlineExit](#) (void)
   
*Command Line Exit.*
- [KZSTATUS\\_t](#) [KzCmdlineAdd](#) (const [KzCmdRegist\\_t](#) \*exe)
   
*Add a Command.*
- [KZSTATUS\\_t](#) [KzCmdlineAddMany](#) (const [KzCmdRegist\\_t](#) \*exes)
   
*Add Many Commands.*
- [KZSTATUS\\_t](#) [KzAddCmdDeviceDriver](#) (void)
   
*Add DeviceDriver commands.*
- [KZSTATUS\\_t](#) [KzSwStart](#) ([KzSwCbk\\_t](#) cbk)
   
*Swith start.*
- [KZSTATUS\\_t](#) [KzSwStop](#) (void)
   
*Swith stop.*
- [KZSTATUS\\_t](#) [KzLedOn](#) ([KzLED\\_t](#) nLedNo)
   
*Set LED ON.*
- [KZSTATUS\\_t](#) [KzLedOff](#) ([KzLED\\_t](#) nLedNo)
   
*Set LED OFF.*

- [KZSTATUS\\_t KzLedBlink](#) ([KzLED\\_t](#) nLedNo, int nOnTime, int nOffTime)  
*Set LED blink.*
- [KZSTATUS\\_t KzMemsStart](#) ([KzMemsCbK\\_t](#) cbk)  
*MEMS start.*
- [KZSTATUS\\_t KzMemsStop](#) (void)  
*MEMS stop.*
- [KZSTATUS\\_t KzAudioStart](#) ([KzAudioCbK\\_t](#) cbk, long lSampleRate, int nBlocks, int nChannels)  
*Audio start.*
- [KZSTATUS\\_t KzAudioStop](#) (void)  
*Audio stop.*
- int [KzGetVersionMajor](#) (void)  
*Get KOBANZAME SDK Major Version.*
- int [KzGetVersionMinor](#) (void)  
*Get KOBANZAME SDK Minor Version.*
- int [KzGetBuildIdx](#) (void)  
*Get KOBANZAME SDK Build index.*
- const char \* [KzGetBuildDate](#) (void)  
*Get KOBANZAME SDK Build Date String.*
- const char \* [KzGetCopyRight](#) (void)  
*Get KOBANZAME SDK Copyright String.*
- [KZSTATUS\\_t KzFilesystemStart](#) (void)  
*Filesystem start.*
- [KZSTATUS\\_t KzAddCmdFilesystem](#) (void)  
*Add Filesystem commands.*
- [KZSTATUS\\_t KzAddCmdStdioTest](#) (void)  
*Add stdio test commands.*

### 7.5.1 Detailed Description

KOBANZAME SDK Standard Include file. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

**Note:**

KOBANZAME SDK Common include file

## 7.5.2 Define Documentation

### 7.5.2.1 #define numof(x) sizeof(x)/sizeof(x[0])

number of the contents

## 7.5.3 Typedef Documentation

### 7.5.3.1 KzAudioCbk\_t

callback function type for audio

**Parameters:**

*in* audio input buffer from Audio Codec

*out* audio output buffer to Audio Codec

*nBlocks* number of the Audio blocks

*nChannels* number of the Audio Channels (1-4)

**Note:**

in and out buffer is clustered as L0/R0/L1/R1

### 7.5.3.2 KzFnCmd\_t

command line function's type

**Parameters:**

*argc* number of the arguments from command line

*argv* argument strings

### 7.5.3.3 KzMemsCbk\_t

callback function type for MEMS changes

**Parameters:**

*x* MEMS X value

*y* MEMS Y value

*z* MEMS Z value



#### 7.5.3.4 KzSwCbk\_t

callback type for physical switches change status

**Parameters:**

*nSw* KOBANZAME hardware switch ID  
*bPushed* TRUE ... Pushed / FALSE ... Released

### 7.5.4 Enumeration Type Documentation

#### 7.5.4.1 enum KzLED\_t

definition for LEDs

**Enumerator:**

*eKzLED0* KOBANZAME hardware LED 0  
*eKzLED1* KOBANZAME hardware LED 1  
*numof\_KzLED* number of the LEDs of KOBANZAME

#### 7.5.4.2 enum KZSTATUS\_t

KOBANZAME SDK's general return type

**Enumerator:**

*KZ\_OK* Success  
*KZ\_ERR* Error

#### 7.5.4.3 enum KzSW\_t

physical switches type

**Enumerator:**

*eKzSW0* KOBANZAME hardware switch 0  
*eKzSW1* KOBANZAME hardware switch 1  
*numof\_KzSW* number of the physical switches

### 7.5.5 Function Documentation

#### 7.5.5.1 KZSTATUS\_t KzAddCmdStdioTest (void)

Add stdio test commands.

**Returns:**

*KZ\_OK* Success  
Lack of memory space for the command. See *KzCmdlineAddMany*

**Note:**

After this function call. User can use following commands.

- puts
- gets

## 7.6 kz\_malloc.c File Reference

re-entrant malloc support `#include <stdlib.h>`

`#include "jsp_kernel.h"`

Include dependency graph for `kz_malloc.c`:

### Defines

- `#define HEAP_CHECK (0)`

### Functions

- `void * malloc (size_t size)`  
*Overload malloc for re-entrant heap operation.*
- `void free (void *ptr)`  
*Overload free for re-entrant heap operation.*

### 7.6.1 Detailed Description

re-entrant malloc support KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### Note:

This file is for supports re-entrant heap operation

## 7.7 kzconfig.h File Reference

KOBANZAME SDK USER configuration file. This graph shows which files directly or indirectly include this file:

### Defines

- #define [KZFILE\\_MAX\\_OPEN](#) (10)  
*Max files that filesystem can open.*
- #define [KZFILE\\_MAX\\_PATH\\_LEN](#) (512)  
*Max file path length.*
- #define [KZCMDLINE\\_MAX\\_INPUT\\_CHAR](#) (128)  
*Max Command Line charactor length.*
- #define [KZCMDLINE\\_MAX\\_COMMANDS](#) (32)  
*Max allocate number of the commands.*
- #define [KZCMDLINE\\_MAX\\_ARGS](#) (16)  
*Max arguments that command line execute function can receive.*
- #define [KZAUDIO\\_MAX\\_AUDIO\\_NBLOCKS](#) (32)  
*Max Audio buffer blocks that callback can receive.*
- #define [KZUSING\\_SD\\_BENCH\\_TEST](#) (1)  
*(1) add SD-Card bench mark test command in filesystem command*

### 7.7.1 Detailed Description

KOBANZAME SDK USER configuration file. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

## 7.8 kzdev.c File Reference

KOBANZAME SDK Device Drivers. `#include "kobanzame.h"`

Include dependency graph for kzdev.c:

### Defines

- `#define USING_LED_AS_DEBUG_PORT` (0)
- `#define KZDEV_SW_NUM` (2)
- `#define KZDEV_SW_SCAN_TIME` (20)
- `#define KZDEV_LED_NUM` (2)
- `#define KZDEV_LED_SCAN_TIME` (5)
- `#define KZDEV_MMC_SCAN_TIME` (10)
- `#define MEMS_SCAN_COUNT` 100

### Enumerations

- enum `LEDStat_t` { `LED_OFF`, `LED_ON`, `LED_Blink` }

### Functions

- void `KzAttIniDeviceDrivers` (ID idCycDevPol, ID idTskDevPol)  
*Driver Initialize.*
- void `KzDevPollingTask` (VP\_INT arg)  
*Device Driver Task.*
- void `KzDevCycHander` (void)  
*Cyclic handler for Device Driver for making time tick.*
- `KZSTATUS_t` `KzSwStart` (`KzSwCbK_t` cbk)  
*Swith start.*
- `KZSTATUS_t` `KzSwStop` (void)  
*Swith stop.*
- `KZSTATUS_t` `KzLedOn` (`KzLED_t` nLedNo)  
*Set LED ON.*
- `KZSTATUS_t` `KzLedOff` (`KzLED_t` nLedNo)  
*Set LED OFF.*
- `KZSTATUS_t` `KzLedBlink` (`KzLED_t` nLedNo, int nOnTime, int nOffTime)  
*Set LED blink.*
- `KZSTATUS_t` `KzMemsStart` (`KzMemsCbK_t` cbk)

*MEMS start.*

- [KZSTATUS\\_t KzMemsStop](#) (void)

*MEMS stop.*

- [KZSTATUS\\_t KzAudioStart](#) ([KzAudioCbk\\_t](#) cbk, long lSampleRate, int nBlocks, int nChannels)

*Audio start.*

- [KZSTATUS\\_t KzAudioStop](#) (void)

*Audio stop.*

- [KZSTATUS\\_t KzAddCmdDeviceDriver](#) (void)

*Add DeviceDriver commands.*

## 7.8.1 Detailed Description

KOBANZAME SDK Device Drivers. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

## 7.8.2 Function Documentation

### 7.8.2.1 void KzAttIniDeviceDrivers (ID *idCycDevPol*, ID *idTskDevPol*)

Driver Initialize.

#### Parameters:

*idCycDevPol* ID for Device driver cyclic hander defined in kernel\_id.h

*idTskDevPol* ID for Device driver task defined in kernel\_id.h

#### Note:

SDK Private function

### 7.8.2.2 void KzDevCycHandler (void)

Cyclic handler for Device Driver for making time tick.

#### Note:

SDK Private function

**7.8.2.3 void KzDevPollingTask (VP\_INT *arg*)**

Device Driver Task.

**Note:**

SDK Private function

## 7.9 kzdev.h File Reference

KOBANZAME SDK Device Driver. #include "kernel.h"

#include "cdefBF533.h"

Include dependency graph for kzdev.h:

This graph shows which files directly or indirectly include this file:

### Defines

- #define **KZ\_GPIO\_SET\_DIR\_OUT**(no) do { \*pFIO\_DIR |= (1<<no); \*pFIO\_INEN &= ~(1<<no); }while(0)
- #define **KZ\_GPIO\_SET\_DIR\_IN**(no) do { \*pFIO\_DIR &= ~(1<<no); \*pFIO\_INEN |= (1<<no); }while(0)
- #define **KZ\_GPIO\_SET\_HI**(no) do { \*pFIO\_FLAG\_S = (1<<no); }while(0)
- #define **KZ\_GPIO\_SET\_LO**(no) do { \*pFIO\_FLAG\_C = (1<<no); }while(0)
- #define **KZ\_GPIO\_SET\_TGL**(no) do { \*pFIO\_FLAG\_T = (1<<no); }while(0)
- #define **KZ\_GPIO\_GET**(no) \*pFIO\_FLAG\_D & ( 1 << no )

### Enumerations

- enum **KZDEV\_SPI\_BITLEN\_t** { **KZDEV\_SPI\_8BIT**, **KZDEV\_SPI\_16BIT** }

### Functions

- int [kzdev\\_spi\\_regist](#) (KZDEV\_SPI\_BITLEN\_t nBitLen, UW dwBps, int nCS, BOOL bCPOL, BOOL bCPHA)  
*SPI Registration.*
- UW [kzdev\\_spi\\_chgbps](#) (int nID, UW dwBps)  
*Change bit ratio.*
- void [kzdev\\_spi\\_open](#) (void)  
*Open SPI ( SPI Semaphore Lock ).*
- void [kzdev\\_spi\\_close](#) (void)  
*Close SPI ( Release SPI Semaphore ).*
- int [kzdev\\_spi\\_readwrite](#) (int nID, const void \*pTx, int nTx, void \*pRx, int nRx)  
*SPI Read and write.*
- int [kzdev\\_spi\\_read](#) (int nID, void \*pRx, int nRx)  
*SPI Read.*



- int `kzdev_spi_write` (int nID, const void \*pTx, int nTx)  
*SPI write.*
- int `kzdev_spi_dma_read` (int nID, void \*pRx, int nRx)  
*SPI DMA Read.*
- int `kzdev_spi_dma_write` (int nID, const void \*pTx, int nTx)  
*SPI DMA write.*
- void `kzdev_mems_att_ini` (void)  
*Initialize MEMS called from uITRON ATT\_INI.*
- BOOL `kzdev_mems_start` (void)  
*Start MEMS.*
- void `kzdev_mems_stop` (void)  
*Stop MEMS.*
- BOOL `kzdev_mems_isReady` (void)  
*Check MEMS data are ready.*
- void `kzdev_mems_getVal` (int \*x, int \*y, int \*z)  
*Get MEMS Value.*
  
- void `kzdev_mmc_att_ini` (void)
- void `disk_timerproc` (void)
- BOOL `kzdev_audio_start` (void \*fnCbk, long lSampleRate, int nBlocks, int nChannels)
- void `kzdev_audio_stop` (void)
- void `kzdev_codec_att_ini` (void)
- void `kzdev_codec_start_DSPMODE` (long lSampleRate)
- void `kzdev_codec_start_I2CMODE` (long lSampleRate)
- void `kzdev_codec_stop` (void)

### 7.9.1 Detailed Description

KOBANZAME SDK Device Driver. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### 7.9.2 Function Documentation

#### 7.9.2.1 void kzdev\_mems\_att\_ini (void)

Initialize MEMS called from uITRON ATT\_INI.

**Note:**

KOBANZAME SDK private function

### 7.9.2.2 void kzdev\_mems\_getVal (int \* x, int \* y, int \* z)

Get MEMS Value.

#### Parameters:

- x* MEMS X value
- y* MEMS Y value
- z* MEMS Z value

#### Note:

KOBANZAME SDK private function

### 7.9.2.3 BOOL kzdev\_mems\_isReady (void)

Check MEMS data are ready.

#### Returns:

TRUE ... Data ready

#### Note:

KOBANZAME SDK private function

### 7.9.2.4 BOOL kzdev\_mems\_start (void)

Start MEMS.

#### Returns:

TRUE ... Success /  
FALSE ... Fail to start (MEMS is broken?)

#### Note:

KOBANZAME SDK private function

### 7.9.2.5 void kzdev\_mems\_stop (void)

Stop MEMS.

#### Note:

KOBANZAME SDK private function

### 7.9.2.6 UW kzdev\_spi\_chgbps (int *nID*, UW *dwBps*)

Change bit ratio.

**Parameters:**

*nID* a SPI ID

*dwBps* bit per second

**Returns:**

Old value

**Note:**

KOBANZAME SDK private function

### 7.9.2.7 void kzdev\_spi\_close (void)

Close SPI ( Release SPI Semaphore ).

**Note:**

KOBANZAME SDK private function

### 7.9.2.8 int kzdev\_spi\_dma\_read (int *nID*, void \* *pRx*, int *nRx*)

SPI DMA Read.

**Parameters:**

*nID* a SPI ID

*pRx* Rx Buffer pointer

*nRx* number of the receive units

**Note:**

KOBANZAME SDK private function

### 7.9.2.9 int kzdev\_spi\_dma\_write (int *nID*, const void \* *pTx*, int *nTx*)

SPI DMA write.

**Parameters:**

*nID* a SPI ID

*pTx* Tx Buffer pointer

*nTx* number of the send units

**Note:**

KOBANZAME SDK private function

**7.9.2.10 void kzdev\_spi\_open (void)**

Open SPI ( SPI Semaphore Lock ).

**Note:**

KOBANZAME SDK private function

**7.9.2.11 int kzdev\_spi\_read (int *nID*, void \* *pRx*, int *nRx*)**

SPI Read.

**Parameters:**

*nID* a SPI ID  
*pRx* Rx Buffer pointer  
*nRx* number of the receive units

**Note:**

KOBANZAME SDK private function

**7.9.2.12 int kzdev\_spi\_readwrite (int *nID*, const void \* *pTx*, int *nTx*, void \* *pRx*, int *nRx*)**

SPI Read and write.

**Parameters:**

*nID* a SPI ID  
*pTx* Tx Buffer pointer  
*nTx* number of the send units  
*pRx* Rx Buffer pointer  
*nRx* number of the receive units

**Note:**

KOBANZAME SDK private function

**7.9.2.13 int kzdev\_spi\_regist (KZDEV\_SPI\_BITLEN\_t *nBitLen*, UW *dwBps*, int *nCS*, BOOL *bCPOL*, BOOL *bCPHA*)**

SPI Registration.

**Parameters:**

*nBitLen* 16bit or 8bit  
*dwBps* bit per second (BPS can be changed by calling `kzdev_spi_chgbps` even after this function is done )  
*nCS* a chip select number , -1 is indicate no chip select work.  
*bCPOL* SPI CPOL is set or not

*bCPHA* SPI CPHA is set or not

**Returns:**

SPI ID

**Note:**

KOBANZAME SDK private function

**7.9.2.14** `int kzdev_spi_write (int nID, const void *pTx, int nTx)`

SPI write.

**Parameters:**

*nID* a SPI ID

*pTx* Tx Buffer pointer

*nTx* number of the send units

**Note:**

KOBANZAME SDK private function

## 7.10 kzdev\_audio.c File Reference

KOBANZAME SDK Audio Driver. #include "kobanzame.h"

#include "jsp\_kernel.h"

Include dependency graph for kzdev\_audio.c:

### Defines

- #define **SAMPLES\_PER\_INTR** (8)
- #define **SLOT\_PER\_SAMPLE** (8)
- #define **INTR\_PER\_BUFFER** (3)
- #define **FLOW\_Autobuffer** (0x1000)
- #define **INTERNAL\_ADC\_L0** (0)
- #define **INTERNAL\_ADC\_L1** (1)
- #define **INTERNAL\_ADC\_R0** (4)
- #define **INTERNAL\_ADC\_R1** (5)
- #define **INTERNAL\_DAC\_L0** (0)
- #define **INTERNAL\_DAC\_L1** (1)
- #define **INTERNAL\_DAC\_L2** (2)
- #define **INTERNAL\_DAC\_R0** (4)
- #define **INTERNAL\_DAC\_R1** (5)
- #define **INTERNAL\_DAC\_R2** (6)
- #define **NUM\_BUF\_AUDIO\_STATE** (2)
- #define **QSHIFT32** (31 - KZQ\_VAL)
- #define **SLEN\_32** 0x001F
- #define **SLEN\_24** 0x0017
- #define **FLOW\_1** 0x1000

### Functions

- void **KzAttIniAudio** (ID idTskDevAudio)
- void **KzAudioISRSport0** (void)
- void **KzAudioDspTask** (VP\_INT arg)
- BOOL **kzdev\_audio\_start** (void \*fnCbk, long lSampleRate, int nBlocks, int nChannel)
- void **kzdev\_audio\_stop** (void)

### Variables

- const long **mSupportedFreq** [] = { 48000 }

### 7.10.1 Detailed Description

KOBANZAME SDK Audio Driver. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

**Author:**

Analog Devices Inc.  
Suikan

**Note:**

This code is copied and modified from VisualDSP++5.0 Audio Codec Talkthrough and AudioFrame-WorkBF533 by Suikan.

### 7.10.2 Define Documentation

#### 7.10.2.1 #define INTR\_PER\_BUFFER (3)

3

#### 7.10.2.2 #define SAMPLES\_PER\_INTR (8)

#### 7.10.2.3 #define SLOT\_PER\_SAMPLE (8)

AD1836A8

## 7.11 kzdev\_codec.c File Reference

KOBANZAME SDK Audio Codec Driver. `#include "kobanzame.h"`

Include dependency graph for `kzdev_codec.c`:

### Defines

- `#define DAC_CONTROL_1 0x0000`
- `#define DAC_CONTROL_2 0x1000`
- `#define DAC_VOLUME_0 0x2000`
- `#define DAC_VOLUME_1 0x3000`
- `#define DAC_VOLUME_2 0x4000`
- `#define DAC_VOLUME_3 0x5000`
- `#define DAC_VOLUME_4 0x6000`
- `#define DAC_VOLUME_5 0x7000`
- `#define ADC_0_PEAK_LEVEL 0x8000`
- `#define ADC_1_PEAK_LEVEL 0x9000`
- `#define ADC_2_PEAK_LEVEL 0xA000`
- `#define ADC_3_PEAK_LEVEL 0xB000`
- `#define ADC_CONTROL_1 0xC000`
- `#define ADC_CONTROL_2 0xD000`
- `#define ADC_CONTROL_3 0xE000`
- `#define TIMOD_DMA_TX 0x0003`

### Functions

- `void kzdev_codec_att_ini (void)`
- `void kzdev_codec_start_DSPMODE (long lSampleRate)`
- `void kzdev_codec_start_I2CMODE (long lSampleRate)`
- `void kzdev_codec_stop (void)`

#### 7.11.1 Detailed Description

KOBANZAME SDK Audio Codec Driver. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.



## 7.12 kzdev\_mems.c File Reference

KOBANZAME SDK MEMS Device Driver. #include "kobanzame.h"

#include "def\_mma74551.h"

Include dependency graph for kzdev\_mems.c:

### Defines

- #define **MESURE\_MODE** (BITATTR\_MMA7455L\_2G | BITATTR\_MMA7455L\_MEASURE\_MODE)
- #define **STANBY\_MODE** (0)
- #define **PORT\_MEMS\_READY** (0)

### Functions

- void [kzdev\\_mems\\_att\\_ini](#) (void)  
*Initialize MEMS called from uITRON ATT\_INI.*
- BOOL [kzdev\\_mems\\_start](#) (void)  
*Start MEMS.*
- void [kzdev\\_mems\\_stop](#) (void)  
*Stop MEMS.*
- BOOL [kzdev\\_mems\\_isReady](#) (void)  
*Check MEMS data are ready.*
- void [kzdev\\_mems\\_getVal](#) (int \*x, int \*y, int \*z)  
*Get MEMS Value.*

### 7.12.1 Detailed Description

KOBANZAME SDK MEMS Device Driver. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### 7.12.2 Function Documentation

#### 7.12.2.1 void kzdev\_mems\_att\_ini (void)

Initialize MEMS called from uITRON ATT\_INI.

**Note:**

KOBANZAME SDK private function

**7.12.2.2 void kzdev\_mems\_getVal (int \* x, int \* y, int \* z)**

Get MEMS Value.

**Parameters:**

*x* MEMS X value

*y* MEMS Y value

*z* MEMS Z value

**Note:**

KOBANZAME SDK private function

**7.12.2.3 BOOL kzdev\_mems\_isReady (void)**

Check MEMS data are ready.

**Returns:**

TRUE ... Data ready

**Note:**

KOBANZAME SDK private function

**7.12.2.4 BOOL kzdev\_mems\_start (void)**

Start MEMS.

**Returns:**

TRUE ... Success /

FALSE ... Fail to start (MEMS is broken?)

**Note:**

KOBANZAME SDK private function

**7.12.2.5 void kzdev\_mems\_stop (void)**

Stop MEMS.

**Note:**

KOBANZAME SDK private function

## 7.13 kzdev\_spi.c File Reference

KOBANZAME SDK SPI Device Driver. #include "kobanzame.h"

#include "jsp\_kernel.h"

Include dependency graph for kzdev\_spi.c:

### Data Structures

- struct [SpiDeviceConfigurator\\_t](#)

### Defines

- #define **MAX\_SPI\_DEVICES** (8)
- #define **SPI\_DEVICE\_SEM** (mSemSpi)
- #define **SPI\_COMPLETE\_SIG** (mSpiCompleteSig)
- #define **SPI\_TIMEOUT\_PER\_BYTE** (10)

### Functions

- void [KzAttIniSpi](#) (ID sem, ID sig)  
*Initialize called from uITRON ATT\_INI.*
- L1CODE void [KzISRSpi](#) (void)  
*SPI Interrupt Service Routine.*
- int [kzdev\\_spi\\_regist](#) (KZDEV\_SPI\_BITLEN\_t nBitLen, UW dwBps, int nCS, BOOL bCPOL, BOOL bCPHA)  
*SPI Registration.*
- UW [kzdev\\_spi\\_chgbps](#) (int nID, UW dwBps)  
*Change bit ratio.*
- void [kzdev\\_spi\\_open](#) (void)  
*Open SPI ( SPI Semaphore Lock ).*
- void [kzdev\\_spi\\_close](#) (void)  
*Close SPI ( Release SPI Semaphore ).*
- int [kzdev\\_spi\\_readwrite](#) (int nID, const void \*pTx, int nTx, void \*pRx, int nRx)  
*SPI Read and write.*
- int [kzdev\\_spi\\_read](#) (int nID, void \*pRx, int nRx)  
*SPI Read.*
- int [kzdev\\_spi\\_write](#) (int nID, const void \*pTx, int nTx)  
*SPI write.*

- int `kzdev_spi_dma_read` (int *nID*, void \**pRx*, int *nRx*)  
*SPI DMA Read.*
- int `kzdev_spi_dma_write` (int *nID*, const void \**pTx*, int *nTx*)  
*SPI DMA write.*

### 7.13.1 Detailed Description

KOBANZAME SDK SPI Device Driver. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### 7.13.2 Function Documentation

#### 7.13.2.1 KzAttIniSpi (ID *sem*, ID *sig*)

Initialize called from uITRON ATT\_INI.

##### Parameters:

- sem* a semaphore ID for re-entrant
- sig* a complete signal

##### Note:

KOBANZAME SDK kernel private function

#### 7.13.2.2 UW `kzdev_spi_chgbps` (int *nID*, UW *dwBps*)

Change bit ratio.

##### Parameters:

- nID* a SPI ID
- dwBps* bit per second

##### Returns:

Old value

##### Note:

KOBANZAME SDK private function

**7.13.2.3 void kzdev\_spi\_close (void)**

Close SPI ( Release SPI Semaphore ).

**Note:**

KOBANZAME SDK private function

**7.13.2.4 int kzdev\_spi\_dma\_read (int *nID*, void \* *pRx*, int *nRx*)**

SPI DMA Read.

**Parameters:**

*nID* a SPI ID

*pRx* Rx Buffer pointer

*nRx* number of the receive units

**Note:**

KOBANZAME SDK private function

**7.13.2.5 int kzdev\_spi\_dma\_write (int *nID*, const void \* *pTx*, int *nTx*)**

SPI DMA write.

**Parameters:**

*nID* a SPI ID

*pTx* Tx Buffer pointer

*nTx* number of the send units

**Note:**

KOBANZAME SDK private function

**7.13.2.6 void kzdev\_spi\_open (void)**

Open SPI ( SPI Semaphore Lock ).

**Note:**

KOBANZAME SDK private function

**7.13.2.7 int kzdev\_spi\_read (int *nID*, void \* *pRx*, int *nRx*)**

SPI Read.

**Parameters:**

*nID* a SPI ID

*pRx* Rx Buffer pointer  
*nRx* number of the receive units

**Note:**

KOBANZAME SDK private function

**7.13.2.8 int kzdev\_spi\_readwrite (int *nID*, const void \* *pTx*, int *nTx*, void \* *pRx*, int *nRx*)**

SPI Read and write.

**Parameters:**

*nID* a SPI ID  
*pTx* Tx Buffer pointer  
*nTx* number of the send units  
*pRx* Rx Buffer pointer  
*nRx* number of the receive units

**Note:**

KOBANZAME SDK private function

**7.13.2.9 int kzdev\_spi\_regist (KZDEV\_SPI\_BITLEN\_t *nBitLen*, UW *dwBps*, int *nCS*, BOOL *bCPOL*, BOOL *bCPHA*)**

SPI Registration.

**Parameters:**

*nBitLen* 16bit or 8bit  
*dwBps* bit per second (BPS can be changed by calling `kzdev_spi_chgbps` even after this function is done )  
*nCS* a chip select number , -1 is indicate no chip select work.  
*bCPOL* SPI CPOL is set or not  
*bCPHA* SPI CPHA is set or not

**Returns:**

SPI ID

**Note:**

KOBANZAME SDK private function

**7.13.2.10 int kzdev\_spi\_write (int *nID*, const void \* *pTx*, int *nTx*)**

SPI write.

**Parameters:**

*nID* a SPI ID

*pTx* Tx Buffer pointer

*nTx* number of the send units

**Note:**

KOBANZAME SDK private function

**7.13.2.11 void KzISRSpI (void)**

SPI Interrupt Service Routine.

**Note:**

KOBANZAME SDK kernel private function

## 7.14 kzdsp.h File Reference

DSP Library. This graph shows which files directly or indirectly include this file:

### Defines

- #define [KZQ\\_VAL](#) (26)

### Functions

- int [KzDspAdd](#) (int a, int b)  
*a + b with saturation*
- int [KzDspSub](#) (int a, int b)  
*a - b with saturation*
- int [KzDspMpy](#) (int a, int b)  
*a \* b with saturation*
- int [KzDspMac](#) (int \*a, int \*b, int n)  
*Multiply and Accumrate.*
- int [KzDspSat](#) (int a)  
*Saturation.*
- int [KzDspAbs](#) (int a)  
*Absolute.*
- int [KzDspShl](#) (int a, int s)  
*Shift left with saturation.*
- int [KzDspShr](#) (int a, int s)  
*Shift right with saturation.*
- int [KzDspNeg](#) (int a)  
*Nagate.*
- float [KzDspTof](#) (int a)  
*To float.*
- int [KzDspToi](#) (float a)  
*To int.*



### 7.14.1 Detailed Description

DSP Libray. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### 7.14.2 Define Documentation

#### 7.14.2.1 #define KZQ\_VAL (26)

KOBANZAME SDK 32bit Q value

## 7.15 kzprivate.h File Reference

SDK Private Global Function define. This graph shows which files directly or indirectly include this file:

### Functions

- void **KzAttIni** (VP\_INT arg)
- void **KzDevCycHander** (void)  
*Cyclic handler for Device Driver for making time tick.*
- void **KzDevPollingTask** (VP\_INT arg)  
*Device Driver Task.*
- void **KzAudioDspTask** (VP\_INT arg)
- void **KzAudioISRSport0** (void)
- const char \* **KzGetCurPath** (void)
- void **KzSetPath** (char \*dest, const char \*arg)  
*SDK Private : Set file path.*
- void **KzSetStdoutFp** (FILE \*fp)  
*Set stdout to fp.*
- void **KzRelStdoutFp** (void)  
*release STDOUT*
- B **KzGetchar** (void)
- void **KzPutchar** (B c)
- int **KzMMCGetCardType** (void)  
*Get SDCARD Type.*

### 7.15.1 Detailed Description

SDK Private Global Function define. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### Note:

ONLY [kobanzame.h](#) can include this code Assume that kobanzame defines are already available.

## 7.15.2 Function Documentation

### 7.15.2.1 void KzDevCycHander (void)

Cyclic handler for Device Driver for making time tick.

**Note:**

SDK Private function

### 7.15.2.2 void KzDevPollingTask (VP\_INT arg)

Device Driver Task.

**Note:**

SDK Private function

### 7.15.2.3 int KzMMCGetCardType (void)

Get SDCARD Type.

**Returns:**

0:MMC 1:SDv1 2:SDv2 3:SDHC

**Note:**

KOBANZAME SDK private function

### 7.15.2.4 void KzSetPath (char \* dest, const char \* arg)

SDK Private : Set file path.

**Parameters:**

*dest* destination buffer

*arg* path argument

## 7.16 kzstdio.c File Reference

KOBANZAME SDK stdio function. #include "kobanzame.h"

```
#include <limits.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

Include dependency graph for kzstdio.c:

### Data Structures

- struct [file\\_t](#)

### Defines

- #define **KZUSING\_STDIO\_TEST** (1)
- #define **\_intptr\_long**
- #define **CONVERT\_BUFLLEN** ((sizeof(\_intptr\_) \* CHAR\_BIT + 2) / 3)

### Functions

- void [KzAttIniStdio](#) (void)  
*Initialize called in uITRON ATT\_INI.*
- void [KzSetStdoutFp](#) (FILE \*fp)  
*Set stdout to fp.*
- void [KzRelStdoutFp](#) (void)  
*release STDOUT*
- int [Kz\\_fgetc](#) (FILE \*fp)  
*Standard C Compatible function: fgetc.*
- int [Kz\\_fputc](#) (int c, FILE \*fp)  
*Standard C Compatible function: fputc.*
- FILE \* [Kz\\_fopen](#) (const char \*filename, const char \*mode)  
*Standard C Compatible function: fopen.*
- int [Kz\\_fclose](#) (FILE \*fp)  
*Standard C Compatible function: fclose.*
- int [Kz\\_fseek](#) (FILE \*fp, long offset, int whence)  
*Standard C Compatible function: fseek.*
- size\_t [Kz\\_fread](#) (void \*ptr, size\_t size, size\_t nmemb, FILE \*stream)

*Standard C Compatible function: fread.*

- `size_t Kz_fwrite` (const void \*ptr, size\_t size, size\_t nmemb, FILE \*stream)  
*Standard C Compatible function: fwrite.*
- `char * Kz_fgets` (char \*s, int n, FILE \*fp)  
*Standard C Compatible function: fgets.*
- `int Kz_fputs` (const char \*s, FILE \*fp)  
*Standard C Compatible function: fputs.*
- `int Kz_fscanf` (FILE \*fp, const char \*format,...)  
*Standard C Compatible function: fscanf.*
- `int Kz_fprintf` (FILE \*fp, const char \*format,...)  
*Standard C Compatible function: fprintf.*
- `int Kz_printf` (const char \*format,...)  
*Standard C Compatible function: printf.*
- `int Kz_scanf` (const char \*format,...)  
*Standard C Compatible function: scanf.*
- `char * Kz_gets` (char \*s)  
*Standard C Compatible function: gets.*
- `int Kz_puts` (const char \*s)  
*Standard C Compatible function: puts.*
- `int Kz_vprintf` (const char \*format, va\_list arg)  
*Standard C Compatible function: vprintf.*
- `int Kz_vfprintf` (FILE \*fp, const char \*format, va\_list arg)  
*Standard C Compatible function: fprintf.*
- `int Kz_getchar` (void)  
*Standard C Compatible function: getchar.*
- `int Kz_putchar` (int c)  
*Standard C Compatible function: putchar.*
- `int Kz_getc` (FILE \*fp)  
*Standard C Compatible function: getc.*
- `int Kz_putc` (int c, FILE \*fp)  
*Standard C Compatible function: putc.*
- `KZSTATUS_t KzAddCmdStdioTest` (void)  
*Add stdio test commands.*

### 7.16.1 Detailed Description

KOBANZAME SDK stdio function. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### 7.16.2 Function Documentation

#### 7.16.2.1 KZSTATUS\_t KzAddCmdStdioTest (void)

Add stdio test commands.

**Returns:**

KZ\_OK Success

Lack of memory space for the command. See KzCmdlineAddMany

**Note:**

After this function call. User can use following commands.

- puts
- gets

## 7.17 kzstdio.h File Reference

KOBANZAME SDK stdio function. `#include <stdarg.h>`

`#include "ff.h"`

Include dependency graph for kzstdio.h:

This graph shows which files directly or indirectly include this file:

### Defines

- `#define FILE` `FIL`
- `#define fopen` `Kz_fopen`
- `#define fclose` `Kz_fclose`
- `#define fseek` `Kz_fseek`
- `#define fread` `Kz_fread`
- `#define fwrite` `Kz_fwrite`
- `#define fgets` `Kz_fgets`
- `#define fputs` `Kz_fputs`
- `#define fgetc` `Kz_fgetc`
- `#define fputc` `Kz_fputc`
- `#define fscanf` `Kz_fscanf`
- `#define fprintf` `Kz_fprintf`
- `#define printf` `Kz_printf`
- `#define scanf` `Kz_scanf`
- `#define getchar` `Kz_getchar`
- `#define putchar` `Kz_putchar`
- `#define gets` `Kz_gets`
- `#define puts` `Kz_puts`
- `#define getc` `Kz_getc`
- `#define putc` `Kz_putc`
- `#define vprintf` `Kz_vprintf`
- `#define vfprintf` `Kz_vfprintf`
- `#define stdin` `((FILE*)-1)`
- `#define stdout` `((FILE*)-1)`
- `#define stderr` `((FILE*)-2)`

### Functions

- `FILE *` [Kz\\_fopen](#) (`const char *filename, const char *mode`)  
*Standard C Compatible function: fopen.*
- `int` [Kz\\_fclose](#) (`FILE *fp`)  
*Standard C Compatible function: fclose.*

- int [Kz\\_fseek](#) (FILE \*fp, long offset, int origin)  
*Standard C Compatible function: fseek.*
- size\_t [Kz\\_fread](#) (void \*ptr, size\_t size, size\_t nmemb, FILE \*stream)  
*Standard C Compatible function: fread.*
- size\_t [Kz\\_fwrite](#) (const void \*ptr, size\_t size, size\_t nmemb, FILE \*stream)  
*Standard C Compatible function: fwrite.*
- char \* [Kz\\_fgets](#) (char \*s, int n, FILE \*fp)  
*Standard C Compatible function: fgets.*
- int [Kz\\_fputs](#) (const char \*s, FILE \*fp)  
*Standard C Compatible function: fputs.*
- int [Kz\\_fgetc](#) (FILE \*fp)  
*Standard C Compatible function: fgetc.*
- int [Kz\\_fputc](#) (int c, FILE \*fp)  
*Standard C Compatible function: fputc.*
- int [Kz\\_fscanf](#) (FILE \*fp, const char \*format,...)  
*Standard C Compatible function: fscanf.*
- int [Kz\\_fprintf](#) (FILE \*fp, const char \*format,...)  
*Standard C Compatible function: fprintf.*
- int [Kz\\_printf](#) (const char \*format,...)  
*Standard C Compatible function: printf.*
- int [Kz\\_scanf](#) (const char \*format,...)  
*Standard C Compatible function: scanf.*
- char \* [Kz\\_gets](#) (char \*s)  
*Standard C Compatible function: gets.*
- int [Kz\\_puts](#) (const char \*s)  
*Standard C Compatible function: puts.*
- int [Kz\\_vprintf](#) (const char \*format, va\_list arg)  
*Standard C Compatible function: vprintf.*
- int [Kz\\_vfprintf](#) (FILE \*fp, const char \*format, va\_list arg)  
*Standard C Compatible function: fprintf.*
- int [Kz\\_getchar](#) (void)  
*Standard C Compatible function: getchar.*
- int [Kz\\_putchar](#) (int c)  
*Standard C Compatible function: putchar.*



- int [Kz\\_getc](#) (FILE \*fp)  
*Standard C Compatible function: getc.*
- int [Kz\\_putc](#) (int c, FILE \*fp)  
*Standard C Compatible function: putc.*

### 7.17.1 Detailed Description

KOBANZAME SDK stdio function. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### Revision

## 7.18 kzversion.h File Reference

KOBANZAME SDK Version File.

### Defines

- #define **KZVER\_MEJOR** (01)
- #define **KZVER\_MINOR** (10)
- #define **KZVER\_BUILD\_IDX** (6)

### 7.18.1 Detailed Description

KOBANZAME SDK Version File. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### Revision

## 7.19 serial\_io.c File Reference

```
KOBANZAME SDK Serial I/O function. #include "target_def.h"
#include "kernel.h"
#include <stdio.h>
#include "serial.h"
Include dependency graph for serial_io.c:
```

### Functions

- B **KzGetchar** (void)
- void **KzPutchar** (B c)

#### 7.19.1 Detailed Description

KOBANZAME SDK Serial I/O function. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

#### Note:

Serial I/O Definitions

## 7.20 target\_def.h File Reference

KOBANZAME SDK Target selector. This graph shows which files directly or indirectly include this file:

### Defines

- #define **TOOL\_GCC** (0)
- #define **TOOL\_VDSP** (1)
- #define **TOOL\_VS2008** (2)

### 7.20.1 Detailed Description

KOBANZAME SDK Target selector. KOBANZAME SDK Software Developers Kit for Blackfin DSP Evaluation Board(KOBANZAME).

Copyright (C) 2010, KOBANZAME SDK Project, all right reserved

LICENSE: The software is a free and you can use and redistribute it for personal, non-profit or commercial products. Redistributions of source code must retain the above copyright notice. There is no warranty in this software, if you suffer any damages by using the software.

### Revision