

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

kobanzame-sdk

1.40

Generated by Doxygen 1.6.2

Fri Feb 26 16:02:56 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:

file_t	??
KzCmdRegist_t	??
SpiDeviceConfigurator_t	??

Chapter 4

File Index

4.1 File List

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

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

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

5.4.1 Function Documentation

5.4.1.1 int KzDspAbs (int a)

Absolute.

Parameters:

a parameter1

Returns:

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

5.4.1.2 int KzDspAdd (int *a*, int *b*)

$a + b$ with saturation

Parameters:

a parameter1
b parameter2

Returns:

$a + b$ with saturation

5.4.1.3 int KzDspMac (int * *a*, int * *b*, int *n*)

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*)

$a * b$ 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*)

Nagate.

Parameters:

a parameter1

Returns:

-*a*

5.4.1.6 int KzDspSat (int *a*)

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*)

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*)

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*)

$a - b$ with saturation

Parameters:

a parameter1

b parameter2

Returns:

$a - b$ with saturation

5.4.1.10 float KzDspTof (int *a*)

To float.

Parameters:

a parameter1

Returns:

(float)*a*

5.4.1.11 int KzDspToi (float *a*)

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 "kzversion.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](#) __attribute__((__section__(".l1scratch")))
Memory section directive: L1 SCRATCH.
- #define [L1DATA_A](#) __attribute__((__section__(".l1data_a")))
Memory section directive: L1 DATA A.
- #define [L1DATA_B](#) __attribute__((__section__(".l1data_b")))
Memory section directive: L1 DATA B.
- #define [L1CODE](#) __attribute__((__section__(".l1code")))
Memory section directive: L1 CODE.
- #define [L3DATA](#) __attribute__((__section__(".l3data")))
Memory section directive: SDRAM DATA.
- #define [L3CODE](#) __attribute__((__section__(".l3code")))
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.
- Inline int [KzGetVersionMajor](#) (void)
Get KOBANZAME SDK Major Version.
- Inline int [KzGetVersionMinor](#) (void)
Get KOBANZAME SDK Minor Version.
- Inline int [KzGetBuildIdx](#) (void)
Get KOBANZAME SDK Build index.
- Inline const char * [KzGetBuildDate](#) (void)
Get KOBANZAME SDK Build Date String.
- Inline 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 <fract.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 "jspi_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 Libray. `#include <fract.h>`

Include dependency graph for kzdsp.h:

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

Defines

- `#define KZQ_VAL` (26)

Functions

- Inline int `KzDspAdd` (int a, int b)
a + b with saturation
- Inline int `KzDspSub` (int a, int b)
a - b with saturation
- Inline int `KzDspMpy` (int a, int b)
*a * b with saturation*
- Inline int `KzDspMac` (int *a, int *b, int n)
Multiply and Accumrate.
- Inline int `KzDspSat` (int a)
Saturation.
- Inline int `KzDspAbs` (int a)
Absolute.
- Inline int `KzDspShl` (int a, int s)
Shift left with saturation.
- Inline int `KzDspShr` (int a, int s)
Shift right with saturation.
- Inline int `KzDspNeg` (int a)
Nagate.
- Inline float `KzDspTof` (int a)
To float.
- Inline 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_BUFLEN ((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. This graph shows which files directly or indirectly include this file:

Defines

- #define **KZVER_MAJOR** (01)
- #define **KZVER_MINOR** (40)
- #define **KZVER_BUILD_IDX** (8)
- #define **KZVER_COPYRIGHT** "KOBANZAME SDK Project"
- #define **KZVER_BUILD_DATE** "Unknown build date"

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