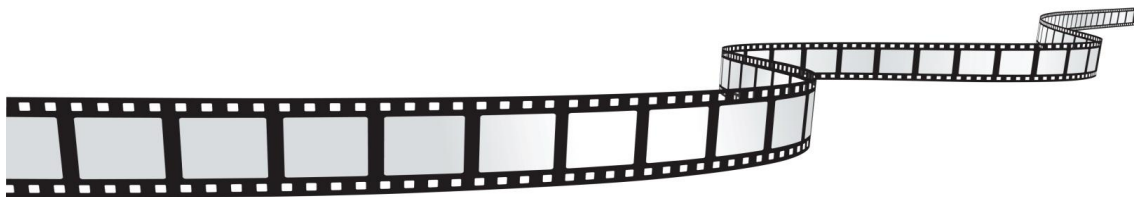


18.08.10



Contents

1	Introduction	4
2	Data Structure Index	5
2.1	Data Structures	5
3	File Index	6
3.1	File List	6
4	Data Structure Documentation	7
4.1	tyAdcChannelConfig Struct Reference	7
4.1.1	Field Documentation	7
4.2	tyAdcContinuousConfig Struct Reference	7
4.2.1	Field Documentation	8
4.3	tyAdcContinuousResult Struct Reference	8
4.3.1	Field Documentation	8
4.4	tyAdcResult Struct Reference	8
4.4.1	Field Documentation	8
4.5	tyAdcResultAllRails Struct Reference	9
4.5.1	Field Documentation	9
4.6	tyBrd198Handle Struct Reference	9
4.6.1	Field Documentation	9
4.7	tyOsBoard0182Configuration Struct Reference	9
4.7.1	Field Documentation	10
4.8	tyRailLookup Struct Reference	10
4.8.1	Field Documentation	10
5	File Documentation	11
5.1	include/brdGpioCfgs/brdMv0182GpioDefaults.h File Reference	11
5.1.1	Detailed Description	11
5.1.2	Variable Documentation	11

5.2	include/brdGpioCfgs/brdMv0182R2GpioDefaults.h File Reference	11
5.2.1	Detailed Description	12
5.2.2	Macro Definition Documentation	12
5.3	include/brdGpioCfgs/brdMv0182R3GpioDefaults.h File Reference	12
5.3.1	Detailed Description	12
5.3.2	Macro Definition Documentation	12
5.4	include/brdGpioCfgs/brdMv0182R4GpioDefaults.h File Reference	13
5.4.1	Detailed Description	13
5.4.2	Macro Definition Documentation	13
5.5	include/brdGpioCfgs/brdMv0182R5GpioDefaults.h File Reference	13
5.5.1	Detailed Description	13
5.5.2	Macro Definition Documentation	14
5.6	include/brdGpioCfgs/brdMv0184GpioDefaults.h File Reference	14
5.6.1	Detailed Description	14
5.6.2	Variable Documentation	14
5.7	include/brdGpioCfgs/brdMv0189GpioDefaults.h File Reference	14
5.7.1	Detailed Description	14
5.7.2	Variable Documentation	15
5.8	include/brdGpioCfgs/brdMv0191GpioDefaults.h File Reference	15
5.8.1	Detailed Description	15
5.8.2	Variable Documentation	15
5.9	include/brdGpioCfgs/brdMv0202GpioDefaults.h File Reference	16
5.9.1	Detailed Description	16
5.9.2	Variable Documentation	16
5.10	include/brdMv0182.h File Reference	16
5.10.1	Detailed Description	17
5.10.2	Function Documentation	17
5.11	include/brdMv0182Defines.h File Reference	18
5.11.1	Detailed Description	19
5.11.2	Macro Definition Documentation	19
5.11.3	Enumeration Type Documentation	21
5.12	include/brdMv0198.h File Reference	21
5.12.1	Detailed Description	22
5.12.2	Function Documentation	22
5.13	include/brdMv0198Defines.h File Reference	26
5.13.1	Detailed Description	28

- 5.13.2 Macro Definition Documentation 28
- 5.13.3 Typedef Documentation 29
- 5.13.4 Enumeration Type Documentation 29
- 5.14 osDrivers/include/OsBrdMv0182.h File Reference 31
 - 5.14.1 Detailed Description 32
 - 5.14.2 Typedef Documentation 32
 - 5.14.3 Enumeration Type Documentation 33
 - 5.14.4 Function Documentation 34
- 5.15 osDrivers/include/OsBrdMv0182Defines.h File Reference 34
- 5.16 intro/MDKdox-BrdDrivers-intro.txt File Reference 34

- Index 35**

Chapter 1

Introduction

This document describes the lower level "Brd" Drivers that are provided for use with Myriad2. The Myriad2 Platform Data Sheet may also be used as a cross reference.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

tyAdcChannelConfig	7
tyAdcContinuousConfig	7
tyAdcContinuousResult	8
tyAdcResult	8
tyAdcResultAllRails	9
tyBrd198Handle	9
tyOsBoard0182Configuration	9
tyRailLookup	10

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

include/ brdMv0182.h	
API for the MV0182 Board Driver	16
include/ brdMv0182Defines.h	
Definitions and types needed by the MV0182 Board Driver API	18
include/ brdMv0198.h	
API for the MV0198 Power Monitor Board Driver	21
include/ brdMv0198Defines.h	
Definitions and types needed by the MV0198 Board Driver API	26
include/brdGpioCfgs/ brdMv0182GpioDefaults.h	
Default GPIO configuration for the MV0182 Board	11
include/brdGpioCfgs/ brdMv0182R2GpioDefaults.h	
Default GPIO configuration for the MV0182 Board	11
include/brdGpioCfgs/ brdMv0182R3GpioDefaults.h	
Default GPIO configuration for the MV0182 Board	12
include/brdGpioCfgs/ brdMv0182R4GpioDefaults.h	
Default GPIO configuration for the MV0182 Board	13
include/brdGpioCfgs/ brdMv0182R5GpioDefaults.h	
Default GPIO configuration for the MV0182 Board	13
include/brdGpioCfgs/ brdMv0184GpioDefaults.h	
Default GPIO configuration for the MV0184 Board	14
include/brdGpioCfgs/ brdMv0189GpioDefaults.h	
Default GPIO configuration for the MV0189 Board	14
include/brdGpioCfgs/ brdMv0191GpioDefaults.h	
Default GPIO configuration for the MV0191 Board	15
include/brdGpioCfgs/ brdMv0202GpioDefaults.h	
Default GPIO configuration for the MV0202 Board	16
osDrivers/include/ OsBrdMv0182.h	
Rtems driver for board mv0182	31
osDrivers/include/ OsBrdMv0182Defines.h	
.	34

Chapter 4

Data Structure Documentation

4.1 tyAdcChannelConfig Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- tyMcpDeviceId [mcpDeviceId](#)
- tyMcpChannel [mcpChannel](#)
- double [channelMultiplier](#)
- double [railNominalV](#)
- tyMcp3424Config * [optMcpChanCfg](#)

4.1.1 Field Documentation

double tyAdcChannelConfig::channelMultiplier

tyMcpChannel tyAdcChannelConfig::mcpChannel

tyMcpDeviceId tyAdcChannelConfig::mcpDeviceId

tyMcp3424Config* tyAdcChannelConfig::optMcpChanCfg

double tyAdcChannelConfig::railNominalV

4.2 tyAdcContinuousConfig Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- tyAdcAChannel [aChannel](#)
- tyAdcBChannel [bChannel](#)
- tyAdcCChannel [cChannel](#)
- tyAdcDChannel [dChannel](#)

4.2.1 Field Documentation

tyAdcAChannel tyAdcContinuousConfig::aChannel

tyAdcBChannel tyAdcContinuousConfig::bChannel

tyAdcCChannel tyAdcContinuousConfig::cChannel

tyAdcDChannel tyAdcContinuousConfig::dChannel

4.3 tyAdcContinuousResult Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- **tyAdcResult** adcAResult
- **tyAdcResult** adcBResult
- **tyAdcResult** adcCResult
- **tyAdcResult** adcDResult

4.3.1 Field Documentation

tyAdcResult tyAdcContinuousResult::adcAResult

tyAdcResult tyAdcContinuousResult::adcBResult

tyAdcResult tyAdcContinuousResult::adcCResult

tyAdcResult tyAdcContinuousResult::adcDResult

4.4 tyAdcResult Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- float **adcValue**
- int **bNewSample**
- **tyChannelId** chanId

4.4.1 Field Documentation

float tyAdcResult::adcValue

int tyAdcResult::bNewSample

tyChannelId tyAdcResult::chanId

4.5 tyAdcResultAllRails Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- float [adcResults](#) [DEFAULT_NUM_RAILS]
- float [totalMilliAmps](#)
- float [totalMilliWatts](#)

4.5.1 Field Documentation

float [tyAdcResultAllRails::adcResults](#)[DEFAULT_NUM_RAILS]

float [tyAdcResultAllRails::totalMilliAmps](#)

float [tyAdcResultAllRails::totalMilliWatts](#)

4.6 tyBrd198Handle Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- I2CM_Device * [i2cDev](#)
- tyMcp3424Handle [mcpHandle](#) [4]
- tyAdcChannelConfig * [adcConfig](#)
- u32 [numAdcChannels](#)
- tyAdcContinuousConfig [continuousCfg](#)

4.6.1 Field Documentation

[tyAdcChannelConfig*](#) [tyBrd198Handle::adcConfig](#)

[tyAdcContinuousConfig](#) [tyBrd198Handle::continuousCfg](#)

I2CM_Device* [tyBrd198Handle::i2cDev](#)

tyMcp3424Handle [tyBrd198Handle::mcpHandle](#)[4]

u32 [tyBrd198Handle::numAdcChannels](#)

4.7 tyOsBoard0182Configuration Struct Reference

```
#include <OsBrdMv0182.h>
```

Data Fields

- `tyOsBoard0182ConfigType` type
- `int` value

4.7.1 Field Documentation

`tyOsBoard0182ConfigType` `tyOsBoard0182Configuration::type`

`int` `tyOsBoard0182Configuration::value`

4.8 tyRailLookup Struct Reference

```
#include <brdMv0198Defines.h>
```

Data Fields

- `u32` `railId`
- `char` `railName` [`MAX_RAIL_STRING_LEN`]

4.8.1 Field Documentation

`u32` `tyRailLookup::railId`

`char` `tyRailLookup::railName`[`MAX_RAIL_STRING_LEN`]

Chapter 5

File Documentation

5.1 `include/brdGpioCfgs/brdMv0182GpioDefaults.h` File Reference

Default GPIO configuration for the MV0182 Board.

```
#include <DrvGpioDefines.h>
```

Variables

- `const drvGpioInitArrayType` [brdMV0182GpioCfgDefault](#)
- `const drvGpioInitArrayType` [brdMV0182R2R3GpioCfgDefault](#)

5.1.1 Detailed Description

Default GPIO configuration for the MV0182 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOs on the MV0182 PCB to good safe initial defaults

5.1.2 Variable Documentation

```
const drvGpioInitArrayType brdMV0182GpioCfgDefault
```

```
const drvGpioInitArrayType brdMV0182R2R3GpioCfgDefault
```

5.2 `include/brdGpioCfgs/brdMv0182R2GpioDefaults.h` File Reference

Default GPIO configuration for the MV0182 Board.

```
#include "brdMv0182GpioDefaults.h"
```

Macros

- `#define brdMV0182R2GpioCfgDefault brdMV0182R2R3GpioCfgDefault`

5.2.1 Detailed Description

Default GPIO configuration for the MV0182 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOs on the MV0182 PCB to good safe initial defaults

5.2.2 Macro Definition Documentation

```
#define brdMV0182R2GpioCfgDefault brdMV0182R2R3GpioCfgDefault
```

5.3 include/brdGpioCfgs/brdMv0182R3GpioDefaults.h File Reference

Default GPIO configuration for the MV0182 Board.

```
#include "brdMv0182GpioDefaults.h"
```

Macros

- `#define brdMV0182R3GpioCfgDefault brdMV0182R2R3GpioCfgDefault`

5.3.1 Detailed Description

Default GPIO configuration for the MV0182 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOs on the MV0182 PCB to good safe initial defaults

5.3.2 Macro Definition Documentation

```
#define brdMV0182R3GpioCfgDefault brdMV0182R2R3GpioCfgDefault
```

5.4 include/brdGpioCfgs/brdMv0182R4GpioDefaults.h File Reference

Default GPIO configuration for the MV0182 Board.

```
#include "brdMv0182GpioDefaults.h"
```

Macros

- #define [brdMV0182R4GpioCfgDefault](#) [brdMV0182R2R3GpioCfgDefault](#)

5.4.1 Detailed Description

Default GPIO configuration for the MV0182 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOS on the MV0182 PCB to good safe initial defaults

5.4.2 Macro Definition Documentation

```
#define brdMV0182R4GpioCfgDefault brdMV0182R2R3GpioCfgDefault
```

5.5 include/brdGpioCfgs/brdMv0182R5GpioDefaults.h File Reference

Default GPIO configuration for the MV0182 Board.

```
#include "brdMv0182GpioDefaults.h"
```

Macros

- #define [brdMV0182R5GpioCfgDefault](#) [brdMV0182R2R3GpioCfgDefault](#)

5.5.1 Detailed Description

Default GPIO configuration for the MV0182 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOS on the MV0182 PCB to good safe initial defaults

5.5.2 Macro Definition Documentation

```
#define brdMV0182R5GpioCfgDefault brdMV0182R2R3GpioCfgDefault
```

5.6 include/brdGpioCfgs/brdMv0184GpioDefaults.h File Reference

Default GPIO configuration for the MV0184 Board.

```
#include <DrvGpioDefines.h>
```

Variables

- const drvGpioInitArrayType **brdMv0184GpioCfgDefault**

5.6.1 Detailed Description

Default GPIO configuration for the MV0184 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialise some of the GPIOs on the MV0184 PCB to good safe initial defaults (to be updated with other GPIOs)

5.6.2 Variable Documentation

```
const drvGpioInitArrayType brdMv0184GpioCfgDefault
```

5.7 include/brdGpioCfgs/brdMv0189GpioDefaults.h File Reference

Default GPIO configuration for the MV0189 Board.

```
#include <DrvGpioDefines.h>
```

Variables

- const drvGpioInitArrayType **brdMv0189GpioCfgDefault**

5.7.1 Detailed Description

Default GPIO configuration for the MV0189 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see-
: common/license.txt

Using the structure defined by this board it is possible to initialise some of the GPIOS on the MV0189 PCB to good safe initial defaults (to be updated with other GPIOS)

5.7.2 Variable Documentation

```
const drvGpioInitArrayType brdMv0189GpioCfgDefault
```

5.8 include/brdGpioCfgs/brdMv0191GpioDefaults.h File Reference

Default GPIO configuration for the MV0191 Board.

```
#include <DrvGpioDefines.h>
```

Variables

- const drvGpioInitArrayType [brdMV0191GpioCfgDefault](#)

5.8.1 Detailed Description

Default GPIO configuration for the MV0191 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see-
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOS on the MV0191 PCB to good safe initial defaults

5.8.2 Variable Documentation

```
const drvGpioInitArrayType brdMV0191GpioCfgDefault
```

Initial value:

```
=
{
    {22, 22 , ACTION_UPDATE_ALL
      ,
      PIN_LEVEL_HIGH
      ,
      D_GPIO_MODE_7          |
      D_GPIO_DIR_OUT        |
      D_GPIO_DATA_INV_OFF   |
      D_GPIO_WAKEUP_OFF
```



```

        ,
        D_GPIO_PAD_DEFAULTS
        , NULL
    },

    {0,0    , ACTION_TERMINATE_ARRAY
        ,
        PIN_LEVEL_LOW
        ,
        D_GPIO_MODE_0
        ,
        D_GPIO_PAD_DEFAULTS
        , NULL
    }
}

```

5.9 include/brdGpioCfgs/brdMv0202GpioDefaults.h File Reference

Default GPIO configuration for the MV0202 Board.

```
#include <DrvGpioDefines.h>
```

Variables

- const drvGpioInitArrayType [brdMV0202GpioCfgDefault](#)

5.9.1 Detailed Description

Default GPIO configuration for the MV0202 Board.

Copyright

All code copyright Movidius Ltd 2014, all rights reserved For License Warranty see:
: common/license.txt

Using the structure defined by this board it is possible to initialize some of the GPIOs on the MV0202 PCB to good safe initial defaults

5.9.2 Variable Documentation

```
const drvGpioInitArrayType brdMV0202GpioCfgDefault
```

5.10 include/brdMv0182.h File Reference

API for the MV0182 Board Driver.

```
#include <brdMv0182Defines.h>
#include <DrvI2cMaster.h>
```

Functions

- s32 [brd182InitialiseI2C](#) (tyI2cConfig *i2c0Cfg, tyI2cConfig *i2c1Cfg, tyI2cConfig *i2c2Cfg, I2CM_Device **i2c0Dev, I2CM_Device **i2c1Dev, I2CM_Device **i2c2Dev)
- s32 [brd182GetPcbRevision](#) (tyMv0182PcbRevision *)
Returns the revision number of the PCB.
- s32 [brd182ExternalPllConfigure](#) (u32 configIndex)
- s32 [brd182SetLed](#) (I2CM_Device *i2cDevice, tyBrdLedId ledNum, tyLedState ledState)

5.10.1 Detailed Description

API for the MV0182 Board Driver.

Copyright

All code copyright Movidius Ltd 2012, all rights reserved For License Warranty see: common/license.txt

5.10.2 Function Documentation

s32 [brd182ExternalPllConfigure](#) (u32 configIndex)

Configures the External PLL to a given frequency

Parameters

in	<i>config_index</i>	(See "DrvCDCEL.h" for usable indexes)
----	---------------------	---------------------------------------

Returns

0 on success, non-zero on fail

s32 [brd182GetPcbRevision](#) (tyMv0182PcbRevision *)

Returns the revision number of the PCB.

s32 [brd182InitialiseI2C](#) (tyI2cConfig * i2c0Cfg, tyI2cConfig * i2c1Cfg, tyI2cConfig * i2c2Cfg, I2CM_Device ** i2c0Dev, I2CM_Device ** i2c1Dev, I2CM_Device ** i2c2Dev)

Initialise the default configuration for I2C0, I2C1, I2C2 on the MV0182 Board

Parameters

in	<i>pointer</i>	to an I2C configuration structure for I2C0 (OR NULL to use board defaults)
----	----------------	--

in	<i>pointer</i>	to an I2C configuration structure for I2C1 (OR NULL to use board defaults)
in	<i>pointer</i>	to an I2C configuration structure for I2C2 (OR NULL to use board defaults)
out	<i>pointer</i>	to storage for an *I2CM_Device Handle for I2C Device 0
out	<i>pointer</i>	to storage for an *I2CM_Device Handle for I2C Device 1
out	<i>pointer</i>	to storage for an *I2CM_Device Handle for I2C Device 2

Returns

0 on Success

`s32 brd182SetLed (I2CM_Device * i2cDevice, tyBrdLedId ledNum, tyLedState ledState)`

Helper function to control LEDS on MV0182

Parameters

in	<i>*I2CM_Device</i>	Handle for I2C Device 2
in	<i>BRD_LED1</i>	or BRD_LED2
in	<i>LED_ON</i>	or LED_OFF

Returns

0 on Success

5.11 include/brdMv0182Defines.h File Reference

Definitions and types needed by the MV0182 Board Driver API.

Macros

- #define `MV0182_OSC_IN_FREQ_KHZ` 12000
- #define `MV0182_I2C0_SCL_PIN` (60)
- #define `MV0182_I2C0_SDA_PIN` (61)
- #define `MV0182_I2C0_SPEED_KHZ_DEFAULT` (100)
- #define `MV0182_I2C0_ADDR_SIZE_DEFAULT` (ADDR_7BIT)
- #define `MV0182_I2C1_SCL_PIN` (12)
- #define `MV0182_I2C1_SDA_PIN` (13)
- #define `MV0182_I2C1_SPEED_KHZ_DEFAULT` (100)
- #define `MV0182_I2C1_ADDR_SIZE_DEFAULT` (ADDR_7BIT)
- #define `MV0182_I2C2_SCL_PIN` (79)
- #define `MV0182_I2C2_SDA_PIN` (80)
- #define `MV0182_I2C2_SPEED_KHZ_DEFAULT` (100)
- #define `MV0182_I2C2_ADDR_SIZE_DEFAULT` (ADDR_7BIT)
- #define `MV0182_WM8325_I2C_ADDR_7BIT` (0x36)
- #define `MV0182_PIN_CAM_A_GPIO0_N` (59)
- #define `MV0182_PIN_CAM_B_GPIO0_N` (15)

- #define MV0182_PIN_COM_IO5_N (56)
- #define MV0182_PIN_AP_IRQ (22)
- #define MV0182_REV_DETECT (9)
- #define CAM_A1_MIPICTRL MIPI_CTRL_0
- #define CAM_B1_MIPICTRL MIPI_CTRL_2
- #define CAM_B2_MIPICTRL MIPI_CTRL_3
- #define CAM_B1_RIGHT_ADDR 0
- #define CAM_B2_LEFT_ADDR 1
- #define CAM_A_ADDR 0
- #define MV0182_MV0201_SENSOR_RST_GPIO MV0182_PIN_CAM_A_GPIO0_N
- #define MV0182_MV0200_SENSOR_LEFT_RST_GPIO MV0182_PIN_COM_IO5_N
- #define MV0182_MV0200_SENSOR_RIGHT_RST_GPIO MV0182_PIN_CAM_B_GPIO0_N
- #define CAMERA_VSYNC_GPIO 0xFF
- #define CAMERA_HSYNC_GPIO 0xFF
- #define CAMERA_VSYNC_GPIO_MODE 0xFF
- #define CAMERA_HSYNC_GPIO_MODE 0xFF

Enumerations

- enum tyMv0182PcbRevision {
MV0182_REV_NOT_INIT = 0, MV0182_R0R1 = 1, MV0182_R2 = 2, MV0182_R3 = 3,
MV0182_R4 = 4, MV0182_R5 = 5 }
- enum tyBrdLedId { BRD_LED1 =1, BRD_LED2 =2 }
- enum tyLedState { LED_ON, LED_OFF }

5.11.1 Detailed Description

Definitions and types needed by the MV0182 Board Driver API.

Copyright

All code copyright Movidius Ltd 2012, all rights reserved For License Warranty see:
: common/license.txt

This header contains all necessary hardware defined constants for this board e.g. GPIO assignments, I2C addresses

5.11.2 Macro Definition Documentation

```
#define CAM_A1_MIPICTRL MIPI_CTRL_0
```

```
#define CAM_A_ADDR 0
```

```
#define CAM_B1_MIPICTRL MIPI_CTRL_2
```

```
#define CAM_B1_RIGHT_ADDR 0
```

```
#define CAM_B2_LEFT_ADDR 1
```

```

#define CAM_B2_MIPICTRL MIPI_CTRL_3

#define CAMERA_HSYNC_GPIO 0xFF

#define CAMERA_HSYNC_GPIO_MODE 0xFF

#define CAMERA_VSYNC_GPIO 0xFF

#define CAMERA_VSYNC_GPIO_MODE 0xFF

#define MV0182_I2C0_ADDR_SIZE_DEFAULT (ADDR_7BIT)

#define MV0182_I2C0_SCL_PIN (60)

#define MV0182_I2C0_SDA_PIN (61)

#define MV0182_I2C0_SPEED_KHZ_DEFAULT (100)

#define MV0182_I2C1_ADDR_SIZE_DEFAULT (ADDR_7BIT)

#define MV0182_I2C1_SCL_PIN (12)

#define MV0182_I2C1_SDA_PIN (13)

#define MV0182_I2C1_SPEED_KHZ_DEFAULT (100)

#define MV0182_I2C2_ADDR_SIZE_DEFAULT (ADDR_7BIT)

#define MV0182_I2C2_SCL_PIN (79)

#define MV0182_I2C2_SDA_PIN (80)

#define MV0182_I2C2_SPEED_KHZ_DEFAULT (100)

#define MV0182_MV0200_SENSOR_LEFT_RST_GPIO MV0182_PIN_COM_IO5_N

#define MV0182_MV0200_SENSOR_RIGHT_RST_GPIO MV0182_PIN_CAM_B_GPIO0_N

#define MV0182_MV0201_SENSOR_RST_GPIO MV0182_PIN_CAM_A_GPIO0_N

#define MV0182_OSC_IN_FREQ_KHZ 12000

#define MV0182_PIN_AP_IRQ (22)

#define MV0182_PIN_CAM_A_GPIO0_N (59)

#define MV0182_PIN_CAM_B_GPIO0_N (15)

#define MV0182_PIN_COM_IO5_N (56)

#define MV0182_REV_DETECT (9)

```

```
#define MV0182_WM8325_I2C_ADDR_7BIT (0x36)
```

5.11.3 Enumeration Type Documentation

enum **tyBrdLedId**

Enumerator

BRD_LED1
BRD_LED2

enum **tyLedState**

Enumerator

LED_ON
LED_OFF

enum **tyMv0182PcbRevision**

Enumerator

MV0182_REV_NOT_INIT
MV0182_R0R1
MV0182_R2
MV0182_R3
MV0182_R4
MV0182_R5

5.12 include/brdMv0198.h File Reference

API for the MV0198 Power Monitor Board Driver.

```
#include "brdMv0198Defines.h"
#include "DrvMcp3424Defines.h"
```

Functions

- int **Brd198Init** (tyBrd198Handle *brdHandle, I2CM_Device *i2cDev, tyAdcChannelConfig *optChannelConfigOverride)
- int **Brd198GetRailValueOneShot** (tyBrd198Handle *brdHandle, tyChannelId chan, float *railValue)
- int **Brd198ConfigureContinuousMode** (tyBrd198Handle *brdHandle, tyAdcContinuousConfig *continuousCfg)
- int **Brd198SampleRailContinuousMode** (tyBrd198Handle *brdHandle, tyAdcContinuousResult *contResult)

- int [Brd198AdjustRailConfig](#) (tyBrd198Handle *brdHandle, tyChannelId chan, tyMcp3424Config *channelConfig)
- int [Brd198LookupRailDescription](#) (tyBrd198Handle *brdHandle, tyChannelId chan, char *channelName, tyChannelType *railType)
- int [Brd198SampleAllRails](#) (tyBrd198Handle *brdHandle, tyAdcResultAllRails *result)
- void [Brd198GetDdrPowerAndCurrent](#) (tyBrd198Handle *brdHandle, tyAdcResultAllRails *samples, float *power, float *current)

5.12.1 Detailed Description

API for the MV0198 Power Monitor Board Driver.

Copyright

All code copyright Movidius Ltd 2012, all rights reserved For License Warranty see: common/license.txt

Allows the user to query the current on any Myriad power rail when used with MV0182

5.12.2 Function Documentation

[int Brd198AdjustRailConfig](#) ([tyBrd198Handle](#) * brdHandle, [tyChannelId](#) chan, [tyMcp3424Config](#) * channelConfig)

Adjust the SampleRate, Gain and Mode of a specific ADC channel

See tyChannelId for the list of supported channels This function takes a pointer to a channelConfig structure which will be used to update the config This function applies to both the OneShot and Continuous Mechanisms however care must be taken to correctly configure the mode paramter accordingly. Note: the channel number within this config will be overwritten with the correct channel number

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
in	<i>ID</i>	of Channel to be read
in	<i>pointer</i>	to MCP channel config structure which is used to modify the configuration

Returns

0 on Success

[int Brd198ConfigureContinuousMode](#) ([tyBrd198Handle](#) * brdHandle, [tyAdcContinuousConfig](#) * continuousCfg)

Configure the MV0198 board in continuous capture mode

Continuous capture mode is an alternative to OneShot mode In this mode up to 4 of ADCs on the platform can be configured to continously capture data. This data can then be polled using the Brd198SampleRail-ContinuousMode API call This function takes a pointer to an [tyAdcContinuousConfig](#) structure which will allows the caller to specify which rail to select on each of the 4 ADCs (A,B,C,D). Optionally the user can also specify that a specific rail is NOT_USED

Parameters

in	<i>Pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
out	<i>Pointer</i>	to tyAdcContinuousConfig structure describing which channels to monitor

Returns

0 on Success

```
void Brd198GetDdrPowerAndCurrent ( tyBrd198Handle * brdHandle, tyAdcResultAllRails *
samples, float * power, float * current )
```

Extract the power and current values only for DDR from a set of samples already taken

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
in	<i>pointer</i>	to a tyAdcResultAllRails structure where the rail values will be found
out	<i>pointer</i>	to the variable where the power value will be returned
out	<i>pointer</i>	to the variable where the current value will be returned

Returns

no return

```
int Brd198GetRailValueOneShot ( tyBrd198Handle * brdHandle, tyChannelId chan, float *
railValue )
```

Retrieve the value of current (or voltage) for a specific Channel

See [tyChannelId](#) for the list of supported channels Most channels return a current in mA, but there are two channels that return a rail voltage (VDDCV_V_MV, MIPI_VDD_V_MV) The function returns a value in either mV or mA depending on the rail e.g. VDDCV_I_MA returns the measured current for the VDDCV rail in mA VDDCV_V_MV returns the measured voltage for the VDDCV rail in mV

Note

This is a blocking function which typically takes 4ms to complete. See ContinuousMode API for an alternative non-blocking mechanism

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
in	<i>ID</i>	of Channel to be read
out	<i>pointer</i>	to storage for result of the measurement

Returns

0 on Success


```
int Brd198Init ( tyBrd198Handle * brdHandle, I2CM_Device * i2cDev, tyAdcChannelConfig *
optChannelConfigOverride )
```

Initialise the MV0198 Power Measurement Board

Passed a handle for a preconfigured I2C device and an optional override for the ADC configuration Table. If the ADC Channel config parameter is NULL, then the default embedded ADC configuration will be used. This Init function is a necessary prerequisite for all the other functions in this API. This API has both blocking and non-blocking methods to query the system power. Specifically the [Brd198GetRailValue-OneShot\(\)](#) call is a blocking function which both initiates a new measurement and doesn't return until the result is available. By default the OneShot mode will take approx 4mS to complete and as such isn't suitable when there are any real-time constraints. The alternative is the ContinuousMode Api which allows the user to poll for new valid data. This API has two methods [Brd198ConfigureContinuousMode](#) to start the measurement and [Brd198SampleRailContinuousMode](#) to check for updated results. Once in continuous mode, the user should not attempt OneShot measurements.

Parameters

out	<i>pointer</i>	to storage for a handle for this MV0198 of type tyBrd198Handle
in	<i>pointer</i>	to a preinitialised I2C device that is connected to the MV0198
in	<i>pointer</i>	to optional override for the default ADC config table, (OR NULL to use board defaults)

Returns

0 on Success

```
int Brd198LookupRailDescription ( tyBrd198Handle * brdHandle, tyChannelId chan, char *
channelName, tyChannelType * railType )
```

Return string containing name of channel and Channel type

This is an optional helper function which allows the user to query the name and type (CURRENT_MEASUREMENT or VOLTAGE_MEASUREMENT) based on the channel ID. The returned string is padded with Spaces to provide alignment.

Note

This function can only be used when no channel override has been provided to [Brd198Init](#)

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
in	<i>ID</i>	of Channel to be queried
out	<i>pointer</i>	to a string (type tyChannelName) which can be used to store the name of the channel
out	<i>pointer</i>	to storage for the type of the measurement (current or voltage)

Returns

0 on Success

```
int Brd198SampleAllRails ( tyBrd198Handle * brdHandle, tyAdcResultAllRails * result )
```

Read all rails once and compute the total number of mA and mW over all rails

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
out	<i>pointer</i>	to a tyAdcResultAllRails structure where the rail values will be found

Returns

0 on Success

```
int Brd198SampleRailContinuousMode ( tyBrd198Handle * brdHandle, tyAdcContinuousResult *
contResult )
```

Query the MV0198 platform for any updated sample results when configured in continuous mode

This function can be called one or more times after `Brd198ConfigureContinuousMode` to check if any ADC has an updated valid sample result. It takes a pointer to storage for a [tyAdcContinuousResult](#) which will be populated with any new data. Specifically for each of the 4 ADCs it has two members, one contains a float representing the data value and the other is a boolean flag which indicates that the sample is valid when TRUE. Most channels return a current, but there are two channels that return a rail voltage (VDDCV_V_MV, MIPI_VDD_V_MV). The function returns a value in either mV or mA depending on the rail e.g. VDDCV_I_MA returns the measured current for the VDDCV rail in mA. VDDCV_V_MV returns the measured voltage for the VDDCV rail in mV.

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0198 of type tyBrd198Handle
out	<i>pointer</i>	to storage for result of the measurement

Returns

0 on Success

5.13 [include/brdMv0198Defines.h](#) File Reference

Definitions and types needed by the MV0198 Board Driver API.

```
#include "mv_types.h"
#include "DrvMcp3424.h"
#include "DrvI2cMaster.h"
#include "DrvCommon.h"
```

Data Structures

- struct [tyRailLookup](#)
- struct [tyAdcResultAllRails](#)
- struct [tyAdcContinuousConfig](#)
- struct [tyAdcResult](#)
- struct [tyAdcContinuousResult](#)
- struct [tyAdcChannelConfig](#)
- struct [tyBrd198Handle](#)

Macros

- #define `MAX_RAIL_STRING_LEN` (20)
- #define `DEFAULT_NUM_RAILS` (16)
- #define `MV0198_MCP_DEV_A_7ADDR` (0x6E)
- #define `MV0198_MCP_DEV_B_7ADDR` (0x6D)
- #define `MV0198_MCP_DEV_C_7ADDR` (0x6A)
- #define `MV0198_MCP_DEV_D_7ADDR` (0x6B)
- #define `VDDCV_I_MUL` (1 / 0.6)
- #define `VDDCC_I_MUL` (1 / 2.0)
- #define `VDDIO_B_I_MUL` (1 / 20.0)
- #define `VDDCR_I_MUL` (1 / 20.0)
- #define `VDDIO_I_MUL` (1 / 10.0)
- #define `MIP_VDD_I_MUL` (1 / 20.0)
- #define `PLL_AVDD_I_MUL` (1 / 20.0)
- #define `DRAM_MVDDQ_I_MUL` (1 / 2.0)
- #define `DRAM_MVDDA_I_MUL` (1 / 20.0)
- #define `DRAM_VDD1_I_MUL` (1 / 20.0)
- #define `DRAM_VDD2_I_MUL` (1 / 10.0)
- #define `DRAM_VDDQ_I_MUL` (1 / 20.0)
- #define `USB_VDD330_I_MUL` (1 / 20.0)
- #define `USB_VP_VDD_I_MUL` (1 / 20.0)
- #define `VDDCV_V_MUL` (1 / 1.0)
- #define `MIPI_VDD_V_MUL` (1 / 1.0)
- #define `CHANNEL_TO_SIGNAL`(chan)

Typedefs

- typedef char `tyChannelName` [`MAX_RAIL_STRING_LEN`]

Enumerations

- enum `tyChannelId` {
`VDDCV_I_MA`, `VDDCC_I_MA`, `VDDIO_B_I_MA`, `VDDCR_I_MA`,
`VDDIO_I_MA`, `MIPI_VDD_I_MA`, `PLL_AVDD_I_MA`, `DRAM_MVDDQ_I_MA`,
`DRAM_MVDDA_I_MA`, `DRAM_VDD1_I_MA`, `DRAM_VDD2_I_MA`, `DRAM_VDDQ_I_MA`,
`USB_VDD330_I_MA`, `USB_VP_VDD_I_MA`, `VDDCV_V_MV`, `MIPI_VDD_V_MV` }
- enum `tyBrt198ErrorCode` {
`DRV_BRD198_DRV_SUCCESS` = `MYR_DRV_SUCCESS`, `DRV_BRD198_DRV_ERROR` =
`MYR_DRV_ERROR`, `DRV_BRD198_DRV_NOT_INITIALIZED` = `MYR_DRV_NOT_INITI-`
`ALIZED`, `DRV_BRD198_DRV_ALREADY_INITIALIZED` = `MYR_DRV_ALREADY_INITI-`
`ALIZED`,
`DRV_BRD198_INVALID_RAIL` = `MYR_DRV_CUSTOM_CODE_START_OFFSET`, `DRV_-`
`BRD198_UNSUPPORTED_ADCCONFG`, `DRV_BRD198_INVALID_CHAN_CONFIG`, `DRV_-`
`BRD198_INVALID_CONT_MODE_NOT_SELECTED` }

- enum `tyAdcAChannel` {
`ADC_A_NOT_USED` = 0, `ADC_A_VDDCR_I_MA` = 1, `ADC_A_VDDCV_I_MA` = 2, `ADC_A_DRAM_VDD1_I_MA` = 3,
`ADC_A_MIPI_VDD_I_MA` = 4 }
- enum `tyAdcBChannel` {
`ADC_B_NOT_USED` = 0, `ADC_B_DRAM_VDD2_I_MA` = 1, `ADC_B_DRAM_VDDQ_I_MA` = 2, `ADC_B_DRAM_MVDDQ_I_MA` = 3,
`ADC_B_DRAM_MVDDA_I_MA` = 4 }
- enum `tyAdcCChannel` {
`ADC_C_NOT_USED` = 0, `ADC_C_USB_VDD330_I_MA` = 1, `ADC_C_USB_VP_VDD_I_MA` = 2, `ADC_C_VDDIO_I_MA` = 3,
`ADC_C_VDD_CC_IOB_MA` = 4 }
- enum `tyAdcDChannel` {
`ADC_D_NOT_USED` = 0, `ADC_D_RESERVED` = 1, `ADC_D_PLL_AVDD_I_MA` = 2, `ADC_D_VDDCV_V_MV` = 3,
`ADC_D_MIPI_VDD_V_MV` = 4 }
- enum `tyChannelType` { `VOLTAGE_MEASUREMENT`, `CURRENT_MEASUREMENT` }

5.13.1 Detailed Description

Definitions and types needed by the MV0198 Board Driver API.

Copyright

All code copyright Movidius Ltd 2012, all rights reserved For License Warranty see:
: common/license.txt

This header contains all necessary hardware defined constants for this board

5.13.2 Macro Definition Documentation

```
#define CHANNEL_TO_SIGNAL( chan )
```

Value:

```
((chan <= VDDCC_I_MA) ? (LOG_EVENT_198_RAIL_BASE + chan) : \
    ((chan == VDDIO_B_I_MA) ? ( \
        LOG_EVENT_198_RAIL_VDDIO_B_I_MUL_I_MA_MA2150) : \
        (LOG_EVENT_198_RAIL_BASE + chan - 1)))
```

```
#define DEFAULT_NUM_RAILS (16)
```

```
#define DRAM_MVDDA_I_MUL (1 / 20.0)
```

```
#define DRAM_MVDDQ_I_MUL (1 / 2.0)
```

```
#define DRAM_VDD1_I_MUL (1 / 20.0)
```

```
#define DRAM_VDD2_I_MUL (1 / 10.0)
```

```
#define DRAM_VDDQ_I_MUL (1 / 20.0)

#define MAX_RAIL_STRING_LEN (20)

#define MIP_VDD_I_MUL (1 / 20.0)

#define MIPI_VDD_V_MUL (1 / 1.0)

#define MV0198_MCP_DEV_A_7ADDR (0x6E)

#define MV0198_MCP_DEV_B_7ADDR (0x6D)

#define MV0198_MCP_DEV_C_7ADDR (0x6A)

#define MV0198_MCP_DEV_D_7ADDR (0x6B)

#define PLL_AVDD_I_MUL (1 / 20.0)

#define USB_VDD330_I_MUL (1 / 20.0)

#define USB_VP_VDD_I_MUL (1 / 20.0)

#define VDDCC_I_MUL (1 / 2.0)

#define VDDCR_I_MUL (1 / 20.0)

#define VDDCV_I_MUL (1 / 0.6)

#define VDDCV_V_MUL (1 / 1.0)

#define VDDIO_B_I_MUL (1 / 20.0)

#define VDDIO_I_MUL (1 / 10.0)
```

5.13.3 Typedef Documentation

```
typedef char tyChannelName[MAX_RAIL_STRING_LEN]
```

5.13.4 Enumeration Type Documentation

```
enum tyAdcAChannel
```

Enumerator

```
ADC_A_NOT_USED
ADC_A_VDDCR_I_MA
ADC_A_VDDCV_I_MA
ADC_A_DRAM_VDD1_I_MA
ADC_A_MIPI_VDD_I_MA
```

enum **tyAdcBChannel**

Enumerator

ADC_B_NOT_USED
ADC_B_DRAM_VDD2_I_MA
ADC_B_DRAM_VDDQ_I_MA
ADC_B_DRAM_MVDDQ_I_MA
ADC_B_DRAM_MVDDA_I_MA

enum **tyAdcCChannel**

Enumerator

ADC_C_NOT_USED
ADC_C_USB_VDD330_I_MA
ADC_C_USB_VP_VDD_I_MA
ADC_C_VDDIO_I_MA
ADC_C_VDD_CC_IOB_MA

enum **tyAdcDChannel**

Enumerator

ADC_D_NOT_USED
ADC_D_RESERVED
ADC_D_PLL_AVDD_I_MA
ADC_D_VDDCV_V_MV
ADC_D_MIPI_VDD_V_MV

enum **tyBrt198ErrorCode**

Enumerator

DRV_BRD198_DRV_SUCCESS
DRV_BRD198_DRV_ERROR
DRV_BRD198_DRV_NOT_INITIALIZED
DRV_BRD198_DRV_ALREADY_INITIALIZED
DRV_BRD198_INVALID_RAIL
DRV_BRD198_UNSUPPORTED_ADCCONFG
DRV_BRD198_INVALID_CHAN_CONFIG
DRV_BRD198_INVALID_CONT_MODE_NOT_SELECTED

enum [tyChannelId](#)

Enumerator

VDDCV_I_MA
VDDCC_I_MA
VDDIO_B_I_MA
VDDCR_I_MA
VDDIO_I_MA
MIPI_VDD_I_MA
PLL_AVDD_I_MA
DRAM_MVDDQ_I_MA
DRAM_MVDDA_I_MA
DRAM_VDD1_I_MA
DRAM_VDD2_I_MA
DRAM_VDDQ_I_MA
USB_VDD330_I_MA
USB_VP_VDD_I_MA
VDDCV_V_MV
MIPI_VDD_V_MV

enum [tyChannelType](#)

Enumerator

VOLTAGE_MEASUREMENT
CURRENT_MEASUREMENT

5.14 [osDrivers/include/OsBrdMv0182.h](#) File Reference

rtems driver for board mv0182

```
#include <mv_types.h>
#include <rtems.h>
#include <OsBmx055.h>
```

Data Structures

- struct [tyOsBoard0182Configuration](#)

Typedefs

- typedef enum [tyOsMv0182I2CConfigType](#) [tyOsMv0182I2CConfigType](#)

- typedef enum `tyOsMv0182CamBBus` `tyOsMv0182CamBBus`
- typedef enum
`tyOsBoard0182ConfigType` `tyOsBoard0182ConfigType`
- typedef struct
`tyOsBoard0182Configuration` `tyOsBoard0182Configuration`

Enumerations

- enum `tyOsBrd182ErrorCode` {
`DRV_OS_BRD_182_DRV_SUCCESS` = `RTEMS_SUCCESSFUL`, `DRV_OS_BRD_182_DRV_ERROR` = `0x8000`, `DRV_OS_BRD_182_I2C_SLAVE_ERROR`, `DRV_OS_BRD_182_I2C_DRIVER_ERROR`,
`DRV_OS_BRD_182_GPIO_DRIVER_ERROR`, `DRV_OS_BRD_182_CPR_INIT_ERROR`, `DRV_OS_BRD_182_CPR_OPEN_ERORR`, `DRV_OS_BRD_182_INVALID_CONFIG`,
`DRV_OS_BRD_182_ALREADY_INITIALIZED`, `DRV_OS_BRD_182_ERROR_BRD_HANDLER_NULL` }
- enum `tyOsMv0182PcbRevision` {
`OS_MV0182_REV_NOT_DETECTED` = 0, `OS_MV0182_R0R1` = 1, `OS_MV0182_R2` = 2, `OS_MV0182_R3` = 3,
`OS_MV0182_R4` = 4, `OS_MV0182_REV_NOT_INIT` = 5, `OS_MV0182_R5` = 6, `OS_MV0212` = 7 }
- enum `tyOsMv0182I2CConfigType` { `MV182_I2C_SLAVE` = 0, `MV182_I2C_MASTER` }
- enum `tyOsMv0182CamBBus` { `MV182_I2C0` = 0, `MV182_I2C1` }
- enum `tyOsBoard0182ConfigType` { `MV182_END` = 0, `MV182_I2C0_STATE`, `MV182_CAM_B_I2C_BUS` }

Functions

- `tyOsBrd182ErrorCode` `osBoard0182Initialise` (`tyOsBoard0182Configuration` *`config`)

5.14.1 Detailed Description

rtems driver for board mv0182

Copyright

All code copyright Movidius Ltd 2014, all rights reserved. For License Warranty see: `common/license.txt`

5.14.2 Typedef Documentation

typedef enum `tyOsBoard0182ConfigType` `tyOsBoard0182ConfigType`

typedef struct `tyOsBoard0182Configuration` `tyOsBoard0182Configuration`

typedef enum `tyOsMv0182CamBBus` `tyOsMv0182CamBBus`

typedef enum `tyOsMv0182I2CConfigType` `tyOsMv0182I2CConfigType`

5.14.3 Enumeration Type Documentation

enum **tyOsBoard0182ConfigType**

Enumerator

MV182_END
MV182_I2C0_STATE
MV182_CAM_B_I2C_BUS

enum **tyOsBrd182ErrorCode**

Enumerator

DRV_OS_BRD_182_DRV_SUCCESS
DRV_OS_BRD_182_DRV_ERROR
DRV_OS_BRD_182_I2C_SLAVE_ERROR
DRV_OS_BRD_182_I2C_DRIVER_ERROR
DRV_OS_BRD_182_GPIO_DRIVER_ERROR
DRV_OS_BRD_182_CPR_INIT_ERROR
DRV_OS_BRD_182_CPR_OPEN_ERORR
DRV_OS_BRD_182_INVALID_CONFIG
DRV_OS_BRD_182_ALREADY_INITIALIZED
DRV_OS_BRD_182_ERROR_BRD_HANDLER_NULL

enum **tyOsMv0182CamBBus**

Enumerator

MV182_I2C0
MV182_I2C1

enum **tyOsMv0182I2CConfigType**

Enumerator

MV182_I2C_SLAVE
MV182_I2C_MASTER

enum **tyOsMv0182PcbRevision**

Enumerator

OS_MV0182_REV_NOT_DETECTED
OS_MV0182_R0R1
OS_MV0182_R2

OS_MV0182_R3
OS_MV0182_R4
OS_MV0182_REV_NOT_INIT
OS_MV0182_R5
OS_MV0212

5.14.4 Function Documentation

tyOsBrd182ErrorCode osBoard0182Initialise (**tyOsBoard0182Configuration** * config)

This function initialize the basic functions of MV0182 board: I2C busses, external clock generator and sets up all GPIOs and detect board revision

Parameters

in	<i>pointer</i>	to the preinitialised handle for this MV0182 of type tyOsBoard0182-Configurations.
out	<i>pointer</i>	to storage the board handler: revision, i2c busses and gpio driver major

5.15 osDrivers/include/OsBrdMv0182Defines.h File Reference

5.16 intro/MDKdox-BrdDrivers-intro.txt File Reference

Index

ADC_A_DRAM_VDD1_I_MA
 brdMv0198Defines.h, [29](#)
 ADC_A_MIPI_VDD_I_MA
 brdMv0198Defines.h, [29](#)
 ADC_A_NOT_USED
 brdMv0198Defines.h, [29](#)
 ADC_A_VDDCR_I_MA
 brdMv0198Defines.h, [29](#)
 ADC_A_VDDCV_I_MA
 brdMv0198Defines.h, [29](#)
 ADC_B_DRAM_MVDDA_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_B_DRAM_MVDDQ_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_B_DRAM_VDD2_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_B_DRAM_VDDQ_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_B_NOT_USED
 brdMv0198Defines.h, [30](#)
 ADC_C_NOT_USED
 brdMv0198Defines.h, [30](#)
 ADC_C_USB_VDD330_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_C_USB_VP_VDD_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_C_VDD_CC_IOB_MA
 brdMv0198Defines.h, [30](#)
 ADC_C_VDDIO_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_D_MIPI_VDD_V_MV
 brdMv0198Defines.h, [30](#)
 ADC_D_NOT_USED
 brdMv0198Defines.h, [30](#)
 ADC_D_PLL_AVDD_I_MA
 brdMv0198Defines.h, [30](#)
 ADC_D_RESERVED
 brdMv0198Defines.h, [30](#)
 ADC_D_VDDCV_V_MV
 brdMv0198Defines.h, [30](#)
 aChannel
 tyAdcContinuousConfig, [8](#)
 adcAResult
 tyAdcContinuousResult, [8](#)
 adcBResult
 tyAdcContinuousResult, [8](#)
 adcCResult
 tyAdcContinuousResult, [8](#)
 adcConfig
 tyBrd198Handle, [9](#)
 adcDResult
 tyAdcContinuousResult, [8](#)
 adcResults
 tyAdcResultAllRails, [9](#)
 adcValue
 tyAdcResult, [8](#)
 BRD_LED1
 brdMv0182Defines.h, [21](#)
 BRD_LED2
 brdMv0182Defines.h, [21](#)
 bChannel
 tyAdcContinuousConfig, [8](#)
 bNewSample
 tyAdcResult, [8](#)
 brd182ExternalPllConfigure
 brdMv0182.h, [17](#)
 brd182GetPcbRevision
 brdMv0182.h, [17](#)
 brd182InitialiseI2C
 brdMv0182.h, [17](#)
 brd182SetLed
 brdMv0182.h, [18](#)
 Brd198AdjustRailConfig
 brdMv0198.h, [22](#)
 Brd198ConfigureContinuousMode
 brdMv0198.h, [22](#)
 Brd198GetDdrPowerAndCurrent
 brdMv0198.h, [23](#)
 Brd198GetRailValueOneShot
 brdMv0198.h, [23](#)
 Brd198Init
 brdMv0198.h, [23](#)
 Brd198LookupRailDescription
 brdMv0198.h, [24](#)
 Brd198SampleAllRails
 brdMv0198.h, [24](#)

Brd198SampleRailContinuousMode
 brdMv0198.h, [26](#)
 brdMv0182Defines.h
 BRD_LED1, [21](#)
 BRD_LED2, [21](#)
 LED_OFF, [21](#)
 LED_ON, [21](#)
 MV0182_R0R1, [21](#)
 MV0182_R2, [21](#)
 MV0182_R3, [21](#)
 MV0182_R4, [21](#)
 MV0182_R5, [21](#)
 MV0182_REV_NOT_INIT, [21](#)
 brdMv0198Defines.h
 ADC_A_DRAM_VDD1_I_MA, [29](#)
 ADC_A_MIPI_VDD_I_MA, [29](#)
 ADC_A_NOT_USED, [29](#)
 ADC_A_VDDCR_I_MA, [29](#)
 ADC_A_VDDCV_I_MA, [29](#)
 ADC_B_DRAM_MVDDA_I_MA, [30](#)
 ADC_B_DRAM_MVDDQ_I_MA, [30](#)
 ADC_B_DRAM_VDD2_I_MA, [30](#)
 ADC_B_DRAM_VDDQ_I_MA, [30](#)
 ADC_B_NOT_USED, [30](#)
 ADC_C_NOT_USED, [30](#)
 ADC_C_USB_VDD330_I_MA, [30](#)
 ADC_C_USB_VP_VDD_I_MA, [30](#)
 ADC_C_VDD_CC_IOB_MA, [30](#)
 ADC_C_VDDIO_I_MA, [30](#)
 ADC_D_MIPI_VDD_V_MV, [30](#)
 ADC_D_NOT_USED, [30](#)
 ADC_D_PLL_AVDD_I_MA, [30](#)
 ADC_D_RESERVED, [30](#)
 ADC_D_VDDCV_V_MV, [30](#)
 CURRENT_MEASUREMENT, [31](#)
 DRAM_MVDDA_I_MA, [31](#)
 DRAM_MVDDQ_I_MA, [31](#)
 DRAM_VDD1_I_MA, [31](#)
 DRAM_VDD2_I_MA, [31](#)
 DRAM_VDDQ_I_MA, [31](#)
 DRV_BRD198_DRV_ALREADY_INITIALIZED, [30](#)
 DRV_BRD198_DRV_ERROR, [30](#)
 DRV_BRD198_DRV_NOT_INITIALIZED, [30](#)
 DRV_BRD198_DRV_SUCCESS, [30](#)
 DRV_BRD198_INVALID_CHAN_CONFIG, [30](#)
 DRV_BRD198_INVALID_CONT_MODE_NOT_SELECTED, [30](#)
 DRV_BRD198_INVALID_RAIL, [30](#)
 DRV_BRD198_UNSUPPORTED_ADCCONFG, [30](#)
 MIPI_VDD_I_MA, [31](#)
 MIPI_VDD_V_MV, [31](#)
 PLL_AVDD_I_MA, [31](#)
 USB_VDD330_I_MA, [31](#)
 USB_VP_VDD_I_MA, [31](#)
 VDDCC_I_MA, [31](#)
 VDDCR_I_MA, [31](#)
 VDDCV_I_MA, [31](#)
 VDDCV_V_MV, [31](#)
 VDDIO_B_I_MA, [31](#)
 VDDIO_I_MA, [31](#)
 VOLTAGE_MEASUREMENT, [31](#)
 brdMV0182GpioCfgDefault
 brdMv0182GpioDefaults.h, [11](#)
 brdMV0182R2GpioCfgDefault
 brdMv0182R2GpioDefaults.h, [12](#)
 brdMV0182R2R3GpioCfgDefault
 brdMv0182GpioDefaults.h, [11](#)
 brdMV0182R3GpioCfgDefault
 brdMv0182R3GpioDefaults.h, [12](#)
 brdMV0182R4GpioCfgDefault
 brdMv0182R4GpioDefaults.h, [13](#)
 brdMV0182R5GpioCfgDefault
 brdMv0182R5GpioDefaults.h, [14](#)
 brdMV0191GpioCfgDefault
 brdMv0191GpioDefaults.h, [15](#)
 brdMV0202GpioCfgDefault
 brdMv0202GpioDefaults.h, [16](#)
 brdMv0182.h
 brd182ExternalPllConfigure, [17](#)
 brd182GetPcbRevison, [17](#)
 brd182InitialiseI2C, [17](#)
 brd182SetLed, [18](#)
 brdMv0182Defines.h
 CAM_A1_MIPICTRL, [19](#)
 CAM_A_ADDR, [19](#)
 CAM_B1_MIPICTRL, [19](#)
 CAM_B1_RIGHT_ADDR, [19](#)
 CAM_B2_LEFT_ADDR, [19](#)
 CAM_B2_MIPICTRL, [19](#)
 CAMERA_HSYNC_GPIO, [20](#)
 CAMERA_VSYNC_GPIO, [20](#)
 MV0182_I2C0_SCL_PIN, [20](#)
 MV0182_I2C0_SDA_PIN, [20](#)
 MV0182_I2C1_SCL_PIN, [20](#)
 MV0182_I2C1_SDA_PIN, [20](#)
 MV0182_I2C2_SCL_PIN, [20](#)
 MV0182_I2C2_SDA_PIN, [20](#)
 MV0182_PIN_AP_IRQ, [20](#)

- MV0182_REV_DETECT, [20](#)
- tyBrdLedId, [21](#)
- tyLedState, [21](#)
- tyMv0182PcbRevision, [21](#)
- brdMv0182GpioDefaults.h
 - brdMV0182GpioCfgDefault, [11](#)
 - brdMV0182R2R3GpioCfgDefault, [11](#)
- brdMv0182R2GpioDefaults.h
 - brdMV0182R2GpioCfgDefault, [12](#)
- brdMv0182R3GpioDefaults.h
 - brdMV0182R3GpioCfgDefault, [12](#)
- brdMv0182R4GpioDefaults.h
 - brdMV0182R4GpioCfgDefault, [13](#)
- brdMv0182R5GpioDefaults.h
 - brdMV0182R5GpioCfgDefault, [14](#)
- brdMv0184GpioCfgDefault
 - brdMv0184GpioDefaults.h, [14](#)
- brdMv0184GpioDefaults.h
 - brdMv0184GpioCfgDefault, [14](#)
- brdMv0189GpioCfgDefault
 - brdMv0189GpioDefaults.h, [15](#)
- brdMv0189GpioDefaults.h
 - brdMv0189GpioCfgDefault, [15](#)
- brdMv0191GpioDefaults.h
 - brdMV0191GpioCfgDefault, [15](#)
- brdMv0198.h
 - Brd198AdjustRailConfig, [22](#)
 - Brd198ConfigureContinuousMode, [22](#)
 - Brd198GetDdrPowerAndCurrent, [23](#)
 - Brd198GetRailValueOneShot, [23](#)
 - Brd198Init, [23](#)
 - Brd198LookupRailDescription, [24](#)
 - Brd198SampleAllRails, [24](#)
 - Brd198SampleRailContinuousMode, [26](#)
- brdMv0198Defines.h
 - CHANNEL_TO_SIGNAL, [28](#)
 - DEFAULT_NUM_RAILS, [28](#)
 - DRAM_MVDDA_I_MUL, [28](#)
 - DRAM_MVDDQ_I_MUL, [28](#)
 - DRAM_VDD1_I_MUL, [28](#)
 - DRAM_VDD2_I_MUL, [28](#)
 - DRAM_VDDQ_I_MUL, [28](#)
 - MIP_VDD_I_MUL, [29](#)
 - MIPI_VDD_V_MUL, [29](#)
 - PLL_AVDD_I_MUL, [29](#)
 - tyAdcAChannel, [29](#)
 - tyAdcBChannel, [29](#)
 - tyAdcCChannel, [30](#)
 - tyAdcDChannel, [30](#)
 - tyBrt198ErrorCode, [30](#)
 - tyChannelId, [30](#)
 - tyChannelName, [29](#)
 - tyChannelType, [31](#)
 - USB_VDD330_I_MUL, [29](#)
 - VDDCC_I_MUL, [29](#)
 - VDDCR_I_MUL, [29](#)
 - VDDCV_I_MUL, [29](#)
 - VDDCV_V_MUL, [29](#)
 - VDDIO_B_I_MUL, [29](#)
 - VDDIO_I_MUL, [29](#)
- brdMv0202GpioDefaults.h
 - brdMV0202GpioCfgDefault, [16](#)
- CURRENT_MEASUREMENT
 - brdMv0198Defines.h, [31](#)
- CAM_A1_MIPICTRL
 - brdMv0182Defines.h, [19](#)
- CAM_A_ADDR
 - brdMv0182Defines.h, [19](#)
- CAM_B1_MIPICTRL
 - brdMv0182Defines.h, [19](#)
- CAM_B1_RIGHT_ADDR
 - brdMv0182Defines.h, [19](#)
- CAM_B2_LEFT_ADDR
 - brdMv0182Defines.h, [19](#)
- CAM_B2_MIPICTRL
 - brdMv0182Defines.h, [19](#)
- CAMERA_HSYNC_GPIO
 - brdMv0182Defines.h, [20](#)
- CAMERA_VSYNC_GPIO
 - brdMv0182Defines.h, [20](#)
- cChannel
 - tyAdcContinuousConfig, [8](#)
- CHANNEL_TO_SIGNAL
 - brdMv0198Defines.h, [28](#)
- chanId
 - tyAdcResult, [8](#)
- channelMultiplier
 - tyAdcChannelConfig, [7](#)
- continuousCfg
 - tyBrd198Handle, [9](#)
- DRAM_MVDDA_I_MA
 - brdMv0198Defines.h, [31](#)
- DRAM_MVDDQ_I_MA
 - brdMv0198Defines.h, [31](#)
- DRAM_VDD1_I_MA
 - brdMv0198Defines.h, [31](#)
- DRAM_VDD2_I_MA
 - brdMv0198Defines.h, [31](#)
- DRAM_VDDQ_I_MA
 - brdMv0198Defines.h, [31](#)

DRV_BRD198_DRV_ALREADY_INITIALIZE-	DRAM_VDDQ_I_MUL
D	brdMv0198Defines.h, 28
brdMv0198Defines.h, 30	
DRV_BRD198_DRV_ERROR	i2cDev
brdMv0198Defines.h, 30	tyBrd198Handle, 9
DRV_BRD198_DRV_NOT_INITIALIZED	include/brdGpioCfgs/brdMv0182GpioDefaults.h,
brdMv0198Defines.h, 30	11
DRV_BRD198_DRV_SUCCESS	include/brdGpioCfgs/brdMv0182R2GpioDefaults.-
brdMv0198Defines.h, 30	h, 11
DRV_BRD198_INVALID_CHAN_CONFIG	include/brdGpioCfgs/brdMv0182R3GpioDefaults.-
brdMv0198Defines.h, 30	h, 12
DRV_BRD198_INVALID_CONT_MODE_NO-	include/brdGpioCfgs/brdMv0182R4GpioDefaults.-
T_SELECTED	h, 13
brdMv0198Defines.h, 30	include/brdGpioCfgs/brdMv0182R5GpioDefaults.-
DRV_BRD198_INVALID_RAIL	h, 13
brdMv0198Defines.h, 30	include/brdGpioCfgs/brdMv0184GpioDefaults.h,
DRV_BRD198_UNSUPPORTED_ADCCONFG	14
brdMv0198Defines.h, 30	include/brdGpioCfgs/brdMv0189GpioDefaults.h,
DRV_OS_BRD_182_ALREADY_INITIALIZED	14
OsBrdMv0182.h, 33	include/brdGpioCfgs/brdMv0191GpioDefaults.h,
DRV_OS_BRD_182_CPR_INIT_ERROR	15
OsBrdMv0182.h, 33	include/brdGpioCfgs/brdMv0202GpioDefaults.h,
DRV_OS_BRD_182_CPR_OPEN_ERORR	16
OsBrdMv0182.h, 33	include/brdMv0182.h, 16
DRV_OS_BRD_182_DRV_ERROR	include/brdMv0182Defines.h, 18
OsBrdMv0182.h, 33	include/brdMv0198.h, 21
DRV_OS_BRD_182_DRV_SUCCESS	include/brdMv0198Defines.h, 26
OsBrdMv0182.h, 33	intro/MDKdox-BrdDrivers-intro.txt, 34
DRV_OS_BRD_182_ERROR_BRD_HANDLE-	
R_NULL	LED_OFF
OsBrdMv0182.h, 33	brdMv0182Defines.h, 21
DRV_OS_BRD_182_GPIO_DRIVER_ERROR	LED_ON
OsBrdMv0182.h, 33	brdMv0182Defines.h, 21
DRV_OS_BRD_182_I2C_DRIVER_ERROR	
OsBrdMv0182.h, 33	MIPI_VDD_I_MA
DRV_OS_BRD_182_I2C_SLAVE_ERROR	brdMv0198Defines.h, 31
OsBrdMv0182.h, 33	MIPI_VDD_V_MV
DRV_OS_BRD_182_INVALID_CONFIG	brdMv0198Defines.h, 31
OsBrdMv0182.h, 33	MV0182_R0R1
dChannel	brdMv0182Defines.h, 21
tyAdcContinuousConfig, 8	MV0182_R2
DEFAULT_NUM_RAILS	brdMv0182Defines.h, 21
brdMv0198Defines.h, 28	MV0182_R3
DRAM_MVDDA_I_MUL	brdMv0182Defines.h, 21
brdMv0198Defines.h, 28	MV0182_R4
DRAM_MVDDQ_I_MUL	brdMv0182Defines.h, 21
brdMv0198Defines.h, 28	MV0182_R5
DRAM_VDD1_I_MUL	brdMv0182Defines.h, 21
brdMv0198Defines.h, 28	MV0182_REV_NOT_INIT
DRAM_VDD2_I_MUL	brdMv0182Defines.h, 21
brdMv0198Defines.h, 28	MV182_CAM_B_I2C_BUS
	OsBrdMv0182.h, 33

MV182_END
 OsBrdMv0182.h, [33](#)
 MV182_I2C0
 OsBrdMv0182.h, [33](#)
 MV182_I2C0_STATE
 OsBrdMv0182.h, [33](#)
 MV182_I2C1
 OsBrdMv0182.h, [33](#)
 MV182_I2C_MASTER
 OsBrdMv0182.h, [33](#)
 MV182_I2C_SLAVE
 OsBrdMv0182.h, [33](#)
 MIP_VDD_I_MUL
 brdMv0198Defines.h, [29](#)
 MIPI_VDD_V_MUL
 brdMv0198Defines.h, [29](#)
 MV0182_I2C0_SCL_PIN
 brdMv0182Defines.h, [20](#)
 MV0182_I2C0_SDA_PIN
 brdMv0182Defines.h, [20](#)
 MV0182_I2C1_SCL_PIN
 brdMv0182Defines.h, [20](#)
 MV0182_I2C1_SDA_PIN
 brdMv0182Defines.h, [20](#)
 MV0182_I2C2_SCL_PIN
 brdMv0182Defines.h, [20](#)
 MV0182_I2C2_SDA_PIN
 brdMv0182Defines.h, [20](#)
 MV0182_PIN_AP_IRQ
 brdMv0182Defines.h, [20](#)
 MV0182_PIN_COM_IO5_N
 brdMv0182Defines.h, [20](#)
 MV0182_REV_DETECT
 brdMv0182Defines.h, [20](#)
 mcpChannel
 tyAdcChannelConfig, [7](#)
 mcpDeviceId
 tyAdcChannelConfig, [7](#)
 mcpHandle
 tyBrd198Handle, [9](#)

 numAdcChannels
 tyBrd198Handle, [9](#)

 OS_MV0182_R0R1
 OsBrdMv0182.h, [33](#)
 OS_MV0182_R2
 OsBrdMv0182.h, [33](#)
 OS_MV0182_R3
 OsBrdMv0182.h, [33](#)
 OS_MV0182_R4
 OsBrdMv0182.h, [34](#)

 OS_MV0182_R5
 OsBrdMv0182.h, [34](#)
 OS_MV0182_REV_NOT_DETECTED
 OsBrdMv0182.h, [33](#)
 OS_MV0182_REV_NOT_INIT
 OsBrdMv0182.h, [34](#)
 OS_MV0212
 OsBrdMv0182.h, [34](#)
 optMcpChanCfg
 tyAdcChannelConfig, [7](#)
 OsBrdMv0182.h
 DRV_OS_BRD_182_ALREADY_INITIALIZED, [33](#)
 DRV_OS_BRD_182_CPR_INIT_ERROR,
 [33](#)
 DRV_OS_BRD_182_CPR_OPEN_ERORR,
 [33](#)
 DRV_OS_BRD_182_DRV_ERROR, [33](#)
 DRV_OS_BRD_182_DRV_SUCCESS, [33](#)
 DRV_OS_BRD_182_ERROR_BRD_HANDLER_NULL, [33](#)
 DRV_OS_BRD_182_GPIO_DRIVER_ERROR, [33](#)
 DRV_OS_BRD_182_I2C_DRIVER_ERROR, [33](#)
 DRV_OS_BRD_182_I2C_SLAVE_ERROR,
 [33](#)
 DRV_OS_BRD_182_INVALID_CONFIG,
 [33](#)
 MV182_CAM_B_I2C_BUS, [33](#)
 MV182_END, [33](#)
 MV182_I2C0, [33](#)
 MV182_I2C0_STATE, [33](#)
 MV182_I2C1, [33](#)
 MV182_I2C_MASTER, [33](#)
 MV182_I2C_SLAVE, [33](#)
 OS_MV0182_R0R1, [33](#)
 OS_MV0182_R2, [33](#)
 OS_MV0182_R3, [33](#)
 OS_MV0182_R4, [34](#)
 OS_MV0182_R5, [34](#)
 OS_MV0182_REV_NOT_DETECTED, [33](#)
 OS_MV0182_REV_NOT_INIT, [34](#)
 OS_MV0212, [34](#)
 osBoard0182Initialise
 OsBrdMv0182.h, [34](#)
 OsBrdMv0182.h
 osBoard0182Initialise, [34](#)
 tyOsBoard0182ConfigType, [32](#), [33](#)
 tyOsBoard0182Configuration, [32](#)
 tyOsBrd182ErrorCode, [33](#)

- tyOsMv0182CamBBus, [32](#), [33](#)
- tyOsMv0182I2CConfigType, [32](#), [33](#)
- tyOsMv0182PcbRevision, [33](#)
- osDrivers/include/OsBrdMv0182.h, [31](#)
- osDrivers/include/OsBrdMv0182Defines.h, [34](#)
- PLL_AVDD_I_MA
 - brdMv0198Defines.h, [31](#)
- PLL_AVDD_I_MUL
 - brdMv0198Defines.h, [29](#)
- railId
 - tyRailLookup, [10](#)
- railName
 - tyRailLookup, [10](#)
- railNominalV
 - tyAdcChannelConfig, [7](#)
- totalMilliAmps
 - tyAdcResultAllRails, [9](#)
- totalMilliWatts
 - tyAdcResultAllRails, [9](#)
- tyAdcAChannel
 - brdMv0198Defines.h, [29](#)
- tyAdcBChannel
 - brdMv0198Defines.h, [29](#)
- tyAdcCChannel
 - brdMv0198Defines.h, [30](#)
- tyAdcChannelConfig, [7](#)
 - channelMultiplier, [7](#)
 - mcpChannel, [7](#)
 - mcpDeviceId, [7](#)
 - optMcpChanCfg, [7](#)
 - railNominalV, [7](#)
- tyAdcContinuousConfig, [7](#)
 - aChannel, [8](#)
 - bChannel, [8](#)
 - cChannel, [8](#)
 - dChannel, [8](#)
- tyAdcContinuousResult, [8](#)
 - adcAResult, [8](#)
 - adcBResult, [8](#)
 - adcCResult, [8](#)
 - adcDResult, [8](#)
- tyAdcDChannel
 - brdMv0198Defines.h, [30](#)
- tyAdcResult, [8](#)
 - adcValue, [8](#)
 - bNewSample, [8](#)
 - chanId, [8](#)
- tyAdcResultAllRails, [9](#)
 - adcResults, [9](#)
- totalMilliAmps, [9](#)
- totalMilliWatts, [9](#)
- tyBrd198Handle, [9](#)
 - adcConfig, [9](#)
 - continuousCfg, [9](#)
 - i2cDev, [9](#)
 - mcpHandle, [9](#)
 - numAdcChannels, [9](#)
- tyBrdLedId
 - brdMv0182Defines.h, [21](#)
- tyBrt198ErrorCode
 - brdMv0198Defines.h, [30](#)
- tyChannelId
 - brdMv0198Defines.h, [30](#)
- tyChannelName
 - brdMv0198Defines.h, [29](#)
- tyChannelType
 - brdMv0198Defines.h, [31](#)
- tyLedState
 - brdMv0182Defines.h, [21](#)
- tyMv0182PcbRevision
 - brdMv0182Defines.h, [21](#)
- tyOsBoard0182ConfigType
 - OsBrdMv0182.h, [32](#), [33](#)
- tyOsBoard0182Configuration, [9](#)
 - OsBrdMv0182.h, [32](#)
 - type, [10](#)
 - value, [10](#)
- tyOsBrd182ErrorCode
 - OsBrdMv0182.h, [33](#)
- tyOsMv0182CamBBus
 - OsBrdMv0182.h, [32](#), [33](#)
- tyOsMv0182I2CConfigType
 - OsBrdMv0182.h, [32](#), [33](#)
- tyOsMv0182PcbRevision
 - OsBrdMv0182.h, [33](#)
- tyRailLookup, [10](#)
 - railId, [10](#)
 - railName, [10](#)
- type
 - tyOsBoard0182Configuration, [10](#)
- USB_VDD330_I_MA
 - brdMv0198Defines.h, [31](#)
- USB_VP_VDD_I_MA
 - brdMv0198Defines.h, [31](#)
- USB_VDD330_I_MUL
 - brdMv0198Defines.h, [29](#)
- USB_VP_VDD_I_MUL
 - brdMv0198Defines.h, [29](#)
- VDDCC_I_MA

brdMv0198Defines.h, [31](#)
 VDDCR_I_MA
 brdMv0198Defines.h, [31](#)
 VDDCV_I_MA
 brdMv0198Defines.h, [31](#)
 VDDCV_V_MV
 brdMv0198Defines.h, [31](#)
 VDDIO_B_I_MA
 brdMv0198Defines.h, [31](#)
 VDDIO_I_MA
 brdMv0198Defines.h, [31](#)
 VOLTAGE_MEASUREMENT
 brdMv0198Defines.h, [31](#)
 VDDCC_I_MUL
 brdMv0198Defines.h, [29](#)
 VDDCR_I_MUL
 brdMv0198Defines.h, [29](#)
 VDDCV_I_MUL
 brdMv0198Defines.h, [29](#)
 VDDCV_V_MUL
 brdMv0198Defines.h, [29](#)
 VDDIO_B_I_MUL
 brdMv0198Defines.h, [29](#)
 VDDIO_I_MUL
 brdMv0198Defines.h, [29](#)
 value
 tyOsBoard0182Configuration, [10](#)