BitInOutKit.h

Go to the documentation of this file.

```
00001 /*****************************************************************************/
00002 /**
00003  *  @file   BitInOutKit.h
00004  *  @brief  BitInOutKit modules makes a kit of all Bit Input/Output ports
00005  *
00006  *  @version    1.0 - Developed by reusing BitInput and BitOutput module
00007  *  @date       Aug-2011
00008  *
00009  * UNPUBLISHED AND CONFIDENTIAL. COPYRIGHT EATON CORPORATION, CREATED 2010,
00010  * ALL RIGHTS RESERVED.
00011  * NOT TO BE REPRODUCED, DISSEMINATED, TRANSFERRED OR USED WITHOUT THE PRIOR
00012  * WRITTEN CONSENT OF EATON CORPORATION.
00013  *****************************************************************************/
00014 #include "BitInOutConfig.h"
00015 #ifdef BIT_IN_OUT_MODULE_USED
00016
00017 #ifndef _BitInOutKit_h
00018 #define _BitInOutKit_h
00019
00020 #include "IOs.h"
00021 #include "Gpio.h"
00022 #include "Kit.h"
00023 #include "global.h"
00024
00025 /*****************************************************************************/
00026 /**
00027  *  @brief This class makes a List of all Bit Input and Output channels registered
00028  */
00029 class BitInOutKit
00030 {
00031 public:
00032     /*************************************************************************/
00033     /** @brief  This enum collects all user defined IDs registered for different
00034      *          Bit Input/Output channels. These IDs will be used by the application
to
00035      *          access any Bit Input/Output channel operations
00036      */
00037     enum Name
00038     {
00039         //Collect all user defined IDs from the configuration file
00040         #define BIT_IO_GPIO_CONFIG
00041             #define BIT_IO_GPIO(user_defined_ID, channel, active_high) \
00042                     user_defined_ID,
00043                 #include "BitInOutConfig.h"
00044             #undef BIT_IO_GPIO
00045         #undef BIT_IO_GPIO_CONFIG
00046
00047         /** Total No. of Bit Input and Output channels registered */
00048         TOT_BIT_IOS,
00049         /** This ID can be used by modules who want to give options to user
00050          *  to use a BitInout channel or can avoid using it */
00051         BIT_IN_OUT_UNUSED
00052
00053     };
00054     /*************************************************************************/
00055     /** @brief This API provides the singleton object for the BitInOutKit class
00056      *  @return - Singleton object for BitInOutKit class
```

```
00057      */
00058     static BitInOutKit *instance();
00059     /*************************************************************************/
00060     /**
00061      * @brief   APi returns the object for the requested client
00062      * @param   client_id   - BitInOut Client ID
00063      * @return  Requested object address
00064      */
00065     BitInOut* handle (BitInOutKit::Name client_id);
00066 private:
00067     /*************************************************************************/
00068     /** @brief Constructor for the class*/
00069     BitInOutKit();
00070     /*************************************************************************/
00071     /** Create Gpio objects for all channels registered. Gpio object names are
00072      *  created by combining the keyword and user defined id */
00073     #define BIT_IO_GPIO_CONFIG
00074         #define BIT_IO_GPIO(user_defined_ID, channel, active_high) \
00075                 Gpio gpio_##user_defined_ID##_obj_;
00076             #include "BitInOutConfig.h"
00077         #undef BIT_IO_GPIO
00078     #undef BIT_IO_GPIO_CONFIG
00079
00080     /*************************************************************************/
00081     /** @brief  Object array to hold address of all individual Gpio object */
00082     BitInOut* bit_in_out_obj_[TOT_BIT_IOS];
00083
00084     /*************************************************************************/
00085     /** @brief  This is a dummy variable. Only used to avoid compilation
00086      *          error in the initializer list of constructor during
00087      *          collecting data from configuration file */
00088     bool dummy_var;
00089 };
00090
00091 #endif  //#ifdef BIT_IN_OUT_MODULE_USED
00092 #endif  //#ifndef _BitInOutKit_h
```

Generated on Wed Jan 4 2012 20:16:03 for Common_ECU_Core_SW by doxygen 1.7.6.1