```
/**
 *
 * @copyright &copy; 2010 - 2020, Fraunhofer-Gesellschaft zur Foerderung der
 *  angewandten Forschung e.V. All rights reserved.
 *
 * BSD 3-Clause License
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 * 1.  Redistributions of source code must retain the above copyright notice,
 *     this list of conditions and the following disclaimer.
 * 2.  Redistributions in binary form must reproduce the above copyright
 *     notice, this list of conditions and the following disclaimer in the
 *     documentation and/or other materials provided with the distribution.
 * 3.  Neither the name of the copyright holder nor the names of its
 *     contributors may be used to endorse or promote products derived from
 *     this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 *
 * We kindly request you to use one or more of the following phrases to refer
 * to foxBMS in your hardware, software, documentation or advertising
 * materials:
 *
 * &Prime;This product uses parts of foxBMS&reg;&Prime;
 *
 * &Prime;This product includes parts of foxBMS&reg;&Prime;
 *
 * &Prime;This product is derived from foxBMS&reg;&Prime;
 *
 */

/**
 * @file    bms_cfg.h
 * @author  foxBMS Team
 * @date    21.09.2015 (date of creation)
 * @ingroup ENGINE_CONF
 * @prefix  BMS
 *
 * @brief   bms driver configuration header
 */

#ifndef BMS_CFG_H_
```

```
#define BMS_CFG_H_

/*================== Includes =============================================*/
#include "general.h"

#include "contactor.h"

/*================== Macros and Definitions ================================*/

#define BMS_REQ_ID_NOREQ 0

/**
 * @ingroup CONFIG_BMS
 * this is the ID that should be requested via CAN signal to go to STANDBY state (ready, but no contactors closed)
 * \par Type:
 * int
 * \par Default:
 * 8
 * \par Range:
 * 0<=x
*/
#define BMS_REQ_ID_STANDBY 8

/**
 * @ingroup CONFIG_BMS
 * this is the ID that should be requested via CAN signal to go to NORMAL state (contactors closing procedure)
 * \par Type:
 * int
 * \par Default:
 * 3
 * \par Range:
 * 0<=x
*/
#define BMS_REQ_ID_NORMAL 3

/**
 * @ingroup CONFIG_BMS
 * this is the ID that should be requested via CAN signal to go to CHARGE state (contactors closing procedure)
 * \par Type:
 * int
 * \par Default:
 * 4
 * \par Range:
 * 0<=x
*/
#define BMS_REQ_ID_CHARGE 4
```

Add Engine request, which is
#define BMS_REQ_ID_ENGINE 5

```
#define BMS_BAL_NO_REQUEST 0

/**
```

```c
 * @ingroup CONFIG_BMS
 * this is the ID that should be requested via CAN to deactivate balancing with override
 * \par Type:
 * int
 * \par Default:
 * 8
 * \par Range:
 * 0<=x
*/
#define BMS_BAL_INACTIVE_OVERRIDE 1

/**
 * @ingroup CONFIG_BMS
 * this is the ID that should be requested via CAN to go out of override mode
 * \par Type:
 * int
 * \par Default:
 * 8
 * \par Range:
 * 0<=x
*/
#define BMS_BAL_OUT_OF_OVERRIDE 2

/**
 * @ingroup CONFIG_BMS
 * this is the ID that should be requested via CAN to activate balancing with override
 * \par Type:
 * int
 * \par Default:
 * 8
 * \par Range:
 * 0<=x
*/
#define BMS_BAL_ACTIVE_OVERRIDE 3


/**
 * BMS statemachine short time definition in ms
 */
#define BMS_STATEMACH_SHORTTIME_MS 1

/**
 * BMS statemachine medium time definition in ms
 */
#define BMS_STATEMACH_MEDIUMTIME_MS 5

/**
 * BMS statemachine long time definition in ms
 */
#define BMS_STATEMACH_LONGTIME_MS 100

/**
```

```
 * BMS statemachine very long time definition in ms
 */
#define BMS_STATEMACH_VERYLONGTIME_MS 2000


/**
 * @ingroup CONFIG_BMS
 * \par Type:
 * int
 * \par Default:
 * 10
 * \par Range:
 * [5,15]
 * \par Unit:
 * 10*ms
*/
#define BMS_SELFPOWERONCHECK_TIMEOUT    10  /* 100ms */

/**
 * @ingroup CONFIG_BMS
 * \par Type:
 * int
 * \par Default:
 * 10
 * \par Range:
 * [5,15]
 * \par Unit:
 * 10*ms
*/
#define BMS_SELFAWAKECHECK_TIMEOUT      10  /* 100ms */


/**
 * @ingroup CONFIG_BMS
 * \par Type:
 * int
 * \par Default:
 * 50
 * \par Range:
 * [40,60]
 * \par Unit:
 * 10*ms
*/
#define BMS_IDLE_TIMEOUT                500  /* 5s timeout to go to sleep or power off in idle state */

#define BMS_GETSELFCHECK_STATE()            BMS_CHECK_OK            /* function could return: BMS_CHECK_NOT_OK or OK BMS_CHECK_BUSY */
#define BMS_GETPOWERONSELFCHECK_STATE()     BMS_CHECK_OK            /* function could return: BMS_CHECK_NOT_OK or OK BMS_CHECK_BUSY */
#define BMS_CHECKPRECHARGE()                BMS_CheckPrecharge()    /* DIAG_CheckPrecharge() */

/*
```

```
 * Mapping the marcos from the contactor-module to
 * bms-macros.
 */
```

The following are not used and should be commented out and then removed at a later time.

```
#define BMS_ALL_CONTACTORS_OFF()         CONT_SwitchAllContactorsOff()

#define BMS_CONT_MAINMINUS_ON()          CONT_SetContactorState(CONT_MAIN_MINUS, CONT_SWITCH_ON)
#define BMS_CONT_MAINMINUS_OFF()         CONT_SetContactorState(CONT_MAIN_MINUS, CONT_SWITCH_OFF)

#define BMS_CONT_MAINPRECHARGE_ON()      CONT_SetContactorState(CONT_PRECHARGE_PLUS, CONT_SWITCH_ON)
#define BMS_CONT_MAINPRECHARGE_OFF()     CONT_SetContactorState(CONT_PRECHARGE_PLUS, CONT_SWITCH_OFF)

#define BMS_CONT_MAINPLUS_ON()           CONT_SetContactorState(CONT_MAIN_PLUS, CONT_SWITCH_ON)
#define BMS_CONT_MAINPLUS_OFF()          CONT_SetContactorState(CONT_MAIN_PLUS, CONT_SWITCH_OFF)

#if BS_SEPARATE_POWERLINES == 1
#define BMS_CONT_CHARGE_MAINMINUS_ON()          CONT_SetContactorState(CONT_CHARGE_MAIN_MINUS, CONT_SWITCH_ON)
#define BMS_CONT_CHARGE_MAINMINUS_OFF()         CONT_SetContactorState(CONT_CHARGE_MAIN_MINUS, CONT_SWITCH_OFF)

#define BMS_CONT_CHARGE_MAINPRECHARGE_ON()      CONT_SetContactorState(CONT_CHARGE_PRECHARGE_PLUS, CONT_SWITCH_ON)
#define BMS_CONT_CHARGE_MAINPRECHARGE_OFF()     CONT_SetContactorState(CONT_CHARGE_PRECHARGE_PLUS, CONT_SWITCH_OFF)

#define BMS_CONT_CHARGE_MAINPLUS_ON()           CONT_SetContactorState(CONT_CHARGE_MAIN_PLUS, CONT_SWITCH_ON)
#define BMS_CONT_CHARGE_MAINPLUS_OFF()          CONT_SetContactorState(CONT_CHARGE_MAIN_PLUS, CONT_SWITCH_OFF)
#endif  /* BS_SEPARATE_POWERLINES == 1 */

/*=================== Function Prototypes ==================================*/

/*=================== Function Implementations =============================*/

#endif /* BMS_CFG_H_ */
```