```
/**
 *
 * @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    sys_cfg.c
 * @author  foxBMS Team
 * @date    21.09.2015 (date of creation)
 * @ingroup ENGINE_CONF
 * @prefix  SYS
 *
 * @brief   Sys driver configuration
 */

/*================== Includes ============================================*/
```

```c
#include "sys_cfg.h"

#include "can.h"
#include "version.h"
#include "stdlib.h"

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

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

/*================== Function Implementations =========
void SYS_SendBootMessage(uint8_t directTransmission) {
    /* Send CAN boot successful message */
    uint8_t data[8];
    data[0] = (uint8_t)atoi((char*)&ver_sw_validation.Version[0]);          /* SW-Version number: major */
    data[1] = (uint8_t)atoi((char*)&ver_sw_validation.Version[2]);          /* SW-Version number: minor */
    data[2] = (uint8_t)atoi((char*)&ver_sw_validation.Version[4]);          /* SW-Version number: bugfix */
    data[3] = 0;
    data[4] = 0xFF & ver_sw_validation.Checksum_u32;
    data[5] = 0xFF & (ver_sw_validation.Checksum_u32 >> 8);
    data[6] = 0xFF & (ver_sw_validation.Checksum_u32 >> 16);
    data[7] = 0xFF & (ver_sw_validation.Checksum_u32 >> 24);
    if (directTransmission == 0) {
        CAN_Send(CAN_NODE0, 0x101, &data[0], 8, 0);
    } else {
        CAN_TxMsg(CAN_NODE0, 0x101, &data[0], 8, 0);
    }
}
```

Send to buffer

Transmit directly

Defined in version.c

```c
typedef struct {
/*0x00*/    uint64_t Valid_u64;                 /* not used (Applicati
/*0x08*/    uint64_t Invalid_u64;               /* not used (Applicati
/*0x10*/    uint32_t Checksum_u32;              /* checksum for valida
/*0x14*/    uint32_t dummy_14;
/*0x18*/    uint8_t  Version[16];               /*Version: "V0.4.0"*/
/*0x28*/    uint8_t  Project[16];               /*Project: "foxBMS"*/
/*0x38*/    VER_BUILDVARIANT_e  BuildVariant;   /*BuildVariant*/
```

The value in "Version" is defined as follows:
* In version.c, it is initialized as VER_SW_VERSION, which is defined as
* BUILD_VERSION_PRIMARY in version.h, which is defined
* in the wscript file of the root, as shown below.

sys_cfg.h | version.c | version.h | wscript

```
227                 [0].replace(':', '-').split('-'))
228     conf.env.timestamp = utcnow
229
230     conf.define('BUILD_APPNAME_PREFIX', conf.env.appname_prefix)
231     for x in variants:
232         conf.define(
233             ('BUILD_APPNAME_{}'.format(x)).upper(),
234             '{}_{}'.format(conf.env.appname_prefix, x)[:14],
235             comment='Define is trimmed to max. 14 characters'.format(x))
236     conf.define('BUILD_VERSION_PRIMARY', conf.env.version_primary)
237     conf.define('BUILD_VERSION_SECONDARY', conf.env.version_secondary)
```

cansignal_cfg.h | version.h | version.c | cansignal_cfg.c | bms.c | ltc.c | diag_cfg.c

mcu-primary > src > general > version.c > ver_sw_validation

```c
const VER_ValidStruct_s __attribute__((section(".FLASH_HEADERSection"))) ver_sw_validation = {
    /* not used (Application-Validation marker) */
        VER_FLASH_APPL_VALID_MARKER,                    /*!< address 0x00*/
    /* not used (Application-Invalidation marker) */
        VER_FLASH_APPL_NOT_INVALID_MARKER,              /*!< address 0x08*/
    /* 32 bit checksum for validating the Application SW (this macro is generated by post build tool)*/
    0,                                                  /*!< address 0x10*/
    0,                                                  /*!< address 0x14*/

    VER_SW_VERSION,                                     /*!< address 0x18, Application SW Version
```

cansignal_cfg.h | version.h | version.c | cansignal_cfg.c | bms.c

mcu-primary > src > general > version.h > VER_SW_VERSION

```c
 * Use Case:
 * uint8_t  Version[16] example:     "V0.4.0 "   (15 char + '\0')
 */
#define VER_SW_VERSION              BUILD_VERSION_PRIMARY       You, a month ago
```