```
/**
 *
 * @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    cansignal_cfg.h
 * @author  foxBMS Team
 * @date    16.09.2015 (date of creation)
 * @ingroup DRIVERS_CONF
 * @prefix  CANS
 *
 * @brief   Headers for the configuration of the messages and signal settings for the CAN driver
 *
 */

```

```
53  #ifndef CANSIGNAL_CFG_H_
54  #define CANSIGNAL_CFG_H_
55
56  /*================== Includes ============================================*/
57  #include "batterysystem_cfg.h"
58  #include "general.h"
59  #include "can_cfg.h"
60
61  /*================== Macros and Definitions ==============================*/
62
63  /**
64   * Default value send when less voltages are configured than voltages values configured for CAN transmission
65   */
66  #define CAN_DEFAULT_VOLTAGE 3000     3.0 V for the current unit which is 1 mV. We can change the unit to 0.1 mV in the future.
67
68  /**
69   * Default value send when less temperatures are configured than temperatures values configured for CAN transmission
70   */
71  #define CAN_DEFAULT_TEMPERATURE 20
72
73  /**
74   * Default value send when less valid flags are configured than valid flags values configured for CAN transmission
75   */
76  #define CAN_DEFAULT_VALID_FLAG 0
77
78  /**
79   * @ingroup CONFIG_CANSIGNAL
80   * task timeslot where the CANS main function is called. Repetition time of periodic CAN messages must be multiple of
     this
81   * \par Type:
82   * select(3)
83   * \par Default:
84   * 1
85  */
86  /* #define CANS_TICK_MS 1 */
87  #define CANS_TICK_MS 10
88  /* #define CANS_TICK_MS 100 */
89
90
91  /**
92   * symbolic names for TX CAN messages. Every used TX message needs to get an individual message name.
93   */
94  typedef enum {     We need to change the name of each item below to CAN0_MSGIdx_xxx in the future to stress that they are the indexes of the CAN messages.
95      /* Insert here symbolic names for CAN0 messages */
96      CAN0_MSG_SystemState_0,  /*!< BMS general state 0 */
97      CAN0_MSG_SystemState_1,  /*!< BMS general state 1 */
98      CAN0_MSG_SystemState_2,  /*!< BMS general state 2 */     Add contactor state index here.
99      CAN0_MSG_SlaveState_0,    /*!< Slave state 0 */
100     CAN0_MSG_SlaveState_1,    /*!< Slave state 1 */
101
102     CAN0_MSG_RecOperatingCurrent,  /*!< Max allowed charge/discharge current */
103     CAN0_MSG_SOP,  /*!< SOP */
```

```
104         CAN0_MSG_SOC,  /*!< SOC */
105         CAN0_MSG_SOH,  /*!< SOH */
106         CAN0_MSG_SOE,  /*!< SOE */
107         CAN0_MSG_MinMaxCellVolt,  /*!< min/max/mean cell voltages */
108         CAN0_MSG_SOV,  /*!< SOV */
109         CAN0_MSG_MinMaxCellTemp,  /*!< min/max/mean cell temperatures */
110         CAN0_MSG_Tempering,  /*!< Tempering */
111         CAN0_MSG_Insulation,  /*!< Insulation */
112         CAN0_MSG_Power_0,  /*!< Moving average power 1s 5s */
113         CAN0_MSG_Power_1,  /*!< Moving average power 10s 30s */
114         CAN0_MSG_Power_2,  /*!< Moving average power 60s configurable duration */
115         CAN0_MSG_Current_0,  /*!< Moving average current 1s 5s */
116         CAN0_MSG_Current_1,  /*!< Moving average current 10s 30s */
117         CAN0_MSG_Current_2,  /*!< Moving average current 60s configurable duration */
118         CAN0_MSG_PackVoltage,  /*!< Pack voltage */
119 
120         CAN0_MSG_Mod0_Cellvolt_0,  /*!< Module 0 Cell voltages 0-2 */
121         CAN0_MSG_Mod0_Cellvolt_1,  /*!< Module 0 Cell voltages 3-5 */
122         CAN0_MSG_Mod0_Cellvolt_2,  /*!< Module 0 Cell voltages 6-8 */
123         CAN0_MSG_Mod0_Cellvolt_3,  /*!< Module 0 Cell voltages 9-11 */
124         CAN0_MSG_Mod0_Cellvolt_4,  /*!< Module 0 Cell voltages 12-14 */
125         CAN0_MSG_Mod0_Cellvolt_5,  /*!< Module 0 Cell voltages 15-17 */
126         CAN0_MSG_Mod0_Celltemp_0,  /*!< Module 0 Cell temperatures 0-2 */
127         CAN0_MSG_Mod0_Celltemp_1,  /*!< Module 0 Cell temperatures 3-5 */
128         CAN0_MSG_Mod0_Celltemp_2,  /*!< Module 0 Cell temperatures 6-8 */
129         CAN0_MSG_Mod0_Celltemp_3,  /*!< Module 0 Cell temperatures 9-11 */
130 
131         CAN0_MSG_Mod1_Cellvolt_0,  /*!< Module 1 Cell voltages 0-2 */
132         CAN0_MSG_Mod1_Cellvolt_1,  /*!< Module 1 Cell voltages 3-5 */
133         CAN0_MSG_Mod1_Cellvolt_2,  /*!< Module 1 Cell voltages 6-8 */
134         CAN0_MSG_Mod1_Cellvolt_3,  /*!< Module 1 Cell voltages 9-11 */
135         CAN0_MSG_Mod1_Cellvolt_4,  /*!< Module 1 Cell voltages 12-14 */
136         CAN0_MSG_Mod1_Cellvolt_5,  /*!< Module 1 Cell voltages 15-17 */
137         CAN0_MSG_Mod1_Celltemp_0,  /*!< Module 1 Cell temperatures 0-2 */
138         CAN0_MSG_Mod1_Celltemp_1,  /*!< Module 1 Cell temperatures 3-5 */
139         CAN0_MSG_Mod1_Celltemp_2,  /*!< Module 1 Cell temperatures 6-8 */
140         CAN0_MSG_Mod1_Celltemp_3,  /*!< Module 1 Cell temperatures 9-11 */
141 
142         CAN0_MSG_Mod2_Cellvolt_0,  /*!< Module 2 Cell voltages 0-2 */
143         CAN0_MSG_Mod2_Cellvolt_1,  /*!< Module 2 Cell voltages 3-5 */
144         CAN0_MSG_Mod2_Cellvolt_2,  /*!< Module 2 Cell voltages 6-8 */
145         CAN0_MSG_Mod2_Cellvolt_3,  /*!< Module 2 Cell voltages 9-11 */
146         CAN0_MSG_Mod2_Cellvolt_4,  /*!< Module 2 Cell voltages 12-14 */
147         CAN0_MSG_Mod2_Cellvolt_5,  /*!< Module 2 Cell voltages 15-17 */
148         CAN0_MSG_Mod2_Celltemp_0,  /*!< Module 2 Cell temperatures 0-2 */
149         CAN0_MSG_Mod2_Celltemp_1,  /*!< Module 2 Cell temperatures 3-5 */
150         CAN0_MSG_Mod2_Celltemp_2,  /*!< Module 2 Cell temperatures 6-8 */
151         CAN0_MSG_Mod2_Celltemp_3,  /*!< Module 2 Cell temperatures 9-11 */
152 
153         CAN0_MSG_Mod3_Cellvolt_0,  /*!< Module 3 Cell voltages 0-2 */
154         CAN0_MSG_Mod3_Cellvolt_1,  /*!< Module 3 Cell voltages 3-5 */
155         CAN0_MSG_Mod3_Cellvolt_2,  /*!< Module 3 Cell voltages 6-8 */
```

```
156       CAN0_MSG_Mod3_Cellvolt_3,  /*!< Module 3 Cell voltages 9-11 */
157       CAN0_MSG_Mod3_Cellvolt_4,  /*!< Module 3 Cell voltages 12-14 */
158       CAN0_MSG_Mod3_Cellvolt_5,  /*!< Module 3 Cell voltages 15-17 */
159       CAN0_MSG_Mod3_Celltemp_0,  /*!< Module 3 Cell temperatures 0-2 */
160       CAN0_MSG_Mod3_Celltemp_1,  /*!< Module 3 Cell temperatures 3-5 */
161       CAN0_MSG_Mod3_Celltemp_2,  /*!< Module 3 Cell temperatures 6-8 */
162       CAN0_MSG_Mod3_Celltemp_3,  /*!< Module 3 Cell temperatures 9-11 */
163
164       CAN0_MSG_Mod4_Cellvolt_0,  /*!< Module 4 Cell voltages 0-2 */
165       CAN0_MSG_Mod4_Cellvolt_1,  /*!< Module 4 Cell voltages 3-5 */
166       CAN0_MSG_Mod4_Cellvolt_2,  /*!< Module 4 Cell voltages 6-8 */
167       CAN0_MSG_Mod4_Cellvolt_3,  /*!< Module 4 Cell voltages 9-11 */
168       CAN0_MSG_Mod4_Cellvolt_4,  /*!< Module 4 Cell voltages 12-14 */
169       CAN0_MSG_Mod4_Cellvolt_5,  /*!< Module 4 Cell voltages 15-17 */
170       CAN0_MSG_Mod4_Celltemp_0,  /*!< Module 4 Cell temperatures 0-2 */
171       CAN0_MSG_Mod4_Celltemp_1,  /*!< Module 4 Cell temperatures 3-5 */
172       CAN0_MSG_Mod4_Celltemp_2,  /*!< Module 4 Cell temperatures 6-8 */
173       CAN0_MSG_Mod4_Celltemp_3,  /*!< Module 4 Cell temperatures 9-11 */
174
175       CAN0_MSG_Mod5_Cellvolt_0,  /*!< Module 5 Cell voltages 0-2 */
176       CAN0_MSG_Mod5_Cellvolt_1,  /*!< Module 5 Cell voltages 3-5 */
177       CAN0_MSG_Mod5_Cellvolt_2,  /*!< Module 5 Cell voltages 6-8 */
178       CAN0_MSG_Mod5_Cellvolt_3,  /*!< Module 5 Cell voltages 9-11 */
179       CAN0_MSG_Mod5_Cellvolt_4,  /*!< Module 5 Cell voltages 12-14 */
180       CAN0_MSG_Mod5_Cellvolt_5,  /*!< Module 5 Cell voltages 15-17 */
181       CAN0_MSG_Mod5_Celltemp_0,  /*!< Module 5 Cell temperatures 0-2 */
182       CAN0_MSG_Mod5_Celltemp_1,  /*!< Module 5 Cell temperatures 3-5 */
183       CAN0_MSG_Mod5_Celltemp_2,  /*!< Module 5 Cell temperatures 6-8 */
184       CAN0_MSG_Mod5_Celltemp_3,  /*!< Module 5 Cell temperatures 9-11 */
185
186       CAN0_MSG_Mod6_Cellvolt_0,  /*!< Module 6 Cell voltages 0-2 */
187       CAN0_MSG_Mod6_Cellvolt_1,  /*!< Module 6 Cell voltages 3-5 */
188       CAN0_MSG_Mod6_Cellvolt_2,  /*!< Module 6 Cell voltages 6-8 */
189       CAN0_MSG_Mod6_Cellvolt_3,  /*!< Module 6 Cell voltages 9-11 */
190       CAN0_MSG_Mod6_Cellvolt_4,  /*!< Module 6 Cell voltages 12-14 */
191       CAN0_MSG_Mod6_Cellvolt_5,  /*!< Module 6 Cell voltages 15-17 */
192       CAN0_MSG_Mod6_Celltemp_0,  /*!< Module 6 Cell temperatures 0-2 */
193       CAN0_MSG_Mod6_Celltemp_1,  /*!< Module 6 Cell temperatures 3-5 */
194       CAN0_MSG_Mod6_Celltemp_2,  /*!< Module 6 Cell temperatures 6-8 */
195       CAN0_MSG_Mod6_Celltemp_3,  /*!< Module 6 Cell temperatures 9-11 */
196
197       CAN0_MSG_Mod7_Cellvolt_0,  /*!< Module 7 Cell voltages 0-2 */
198       CAN0_MSG_Mod7_Cellvolt_1,  /*!< Module 7 Cell voltages 3-5 */
199       CAN0_MSG_Mod7_Cellvolt_2,  /*!< Module 7 Cell voltages 6-8 */
200       CAN0_MSG_Mod7_Cellvolt_3,  /*!< Module 7 Cell voltages 9-11 */
201       CAN0_MSG_Mod7_Cellvolt_4,  /*!< Module 7 Cell voltages 12-14 */
202       CAN0_MSG_Mod7_Cellvolt_5,  /*!< Module 7 Cell voltages 15-17 */
203       CAN0_MSG_Mod7_Celltemp_0,  /*!< Module 7 Cell temperatures 0-2 */
204       CAN0_MSG_Mod7_Celltemp_1,  /*!< Module 7 Cell temperatures 3-5 */
205       CAN0_MSG_Mod7_Celltemp_2,  /*!< Module 7 Cell temperatures 6-8 */
206       CAN0_MSG_Mod7_Celltemp_3,  /*!< Module 7 Cell temperatures 9-11 */
207
```

```
208  #ifdef CURRENT_SENSOR_ISABELLENHUETTE_TRIGGERED
209      CAN0_MSG_BMS_CurrentTrigger,    /*!< Cell Voltages Max Min Average */
210  #endif /* CURRENT_SENSOR_ISABELLENHUETTE_TRIGGERED */
211
212
213      /* Insert here symbolic names for CAN1 messages */
214  } CANS_messagesTx_e;
215
216  /**
217   * symbolic names for RX CAN messages
218   */
219  typedef enum {          They are indexes as well
220      /* Insert here symbolic names for CAN0 messages */
221      CAN0_MSG_StateRequest,                    /*!< state request */
222      CAN0_MSG_SW_RESET,                        /*!< can message for SW reset */
223      CAN0_MSG_IVT_Current,                     /*!< current sensing */
224      CAN0_MSG_IVT_Voltage_1,                   /*!< current sensor voltage 1 */
225      CAN0_MSG_IVT_Voltage_2,                   /*!< current sensor voltage 2 */
226      CAN0_MSG_IVT_Voltage_3,                   /*!< current sensor voltage 3 */
227      CAN0_MSG_IVT_Temperature,                 /*!< current sensor temperature */
228      CAN0_MSG_IVT_Power,                       /*!< current sensor power */
229      CAN0_MSG_IVT_CoulombCount,                /*!< current sensor C-C */
230      CAN0_MSG_IVT_EnergyCount,                 /*!< current sensor E-C */
231      CAN0_MSG_DEBUG,                           /*!< debug messages */
232      CAN0_MSG_GetReleaseVersion,               /*!< Get SW release version */
233      CAN0_MSG_EngineRequest,
234
235      /* Insert here symbolic names for CAN1 messages */
236  } CANS_messagesRx_e;
237
238  /**
239   * symbolic names for CAN0 transmission signals
240   */
241  typedef enum {      Below are the CAN signal indexes. Since signals are included in messages, there is no confusion and hence we can use
                        CAN0_SIG_xx without adding Idx in the name.
242      CAN0_SIG_GS0_general_error,  /* 0:good, 1:error */
243      CAN0_SIG_GS0_current_state,  /* currently no used */
244      CAN0_SIG_GS0_error_overtemp_charge,  /* 0:good, 1:error */
245      CAN0_SIG_GS0_error_undertemp_charge,  /* 0:good, 1:error */
246      CAN0_SIG_GS0_error_overtemp_discharge,  /* 0:good, 1:error */
247      CAN0_SIG_GS0_error_undertemp_discharge,  /* 0:good, 1:error */
248      CAN0_SIG_GS0_error_overcurrent_charge,  /* 0:good, 1:error */
249      CAN0_SIG_GS0_error_overcurrent_discharge,  /* 0:good, 1:error */
250
251      CAN0_SIG_GS1_error_overvoltage,  /* 0:good, 1:error */
252      CAN0_SIG_GS1_error_undervoltage,  /* 0:good, 1:error */
253      CAN0_SIG_GS1_error_deep_discharge,  /* 0:good, 1:error */
254      CAN0_SIG_GS1_error_temperature_MCU0,  /* 0:good, 1:error */
255      CAN0_SIG_GS1_error_contactor,  /* 0:good, 1:error */
256      CAN0_SIG_GS1_error_selftest,  /* 0:good, 1:error */
257      CAN0_SIG_GS1_error_cantiming,  /* 0:good, 1:error */
258      CAN0_SIG_GS1_current_sensor,  /* 0:good, 1:error */
259      CAN0_SIG_GS1_balancing_active,  /* 0:off, 1:on */
```

```
260
261       CAN0_SIG_GS2_state_cont_interlock,  /* bitfield 0:off, 1:on */
262       CAN0_SIG_GS2_error_insulation,  /* 0: good, 1: error */
263       CAN0_SIG_GS2_fuse_state,  /* 0: fuse intact, 1: fuse tripped */
264       CAN0_SIG_GS2_lowCoinCellVolt,  /* 0: okay, else: low voltage */
265       CAN0_SIG_GS2_error_openWire,  /* 0: okay, 1: open wire detected */
266       CAN0_SIG_GS2_daisyChain,  /* 0: okay, else: error */
267       CAN0_SIG_GS2_plausibilityCheck,  /* 0: okay, else: error */
268
269       CAN0_SIG_SS0_states,  /* 0: good, 1: error */
270       CAN0_SIG_SS1_states,  /* 0: good, 1: error */
271
272       CAN0_SIG_RecChargeCurrent,
273       CAN0_SIG_RecChargeCurrent_Peak,
274       CAN0_SIG_RecDischargeCurrent,
275       CAN0_SIG_RecDischargeCurrent_Peak,
276
277       CAN0_SIG_MaxChargePower,
278       CAN0_SIG_MaxChargePower_Peak,
279       CAN0_SIG_MaxDischargePower,
280       CAN0_SIG_MaxDischargePower_Peak,
281
282       CAN0_SIG_SOC_mean,
283       CAN0_SIG_SOC_min,
284       CAN0_SIG_SOC_max,
285
286       CAN0_SIG_SOH_mean,
287       CAN0_SIG_SOH_min,
288       CAN0_SIG_SOH_max,
289
290       CAN0_SIG_SOE,
291       CAN0_SIG_RemainingEnergy,
292
293       CAN0_SIG_Cellvolt_mean,
294       CAN0_SIG_Cellvolt_min,
295       CAN0_SIG_Cellvolt_max,
296       CAN0_SIG_ModNumber_volt_min,
297       CAN0_SIG_ModNumber_volt_max,
298
299       CAN0_SIG_SOV,
300
301       CAN0_SIG_Celltemp_mean,
302       CAN0_SIG_Celltemp_min,
303       CAN0_SIG_Celltemp_max,
304       CAN0_SIG_ModNumber_temp_min,
305       CAN0_SIG_ModNumber_temp_max,
306
307       CAN0_SIG_CoolingNeeded,
308       CAN0_SIG_HeatingNeeded,
309       CAN0_SIG_TemperingDemand,
310
311       CAN0_SIG_InsulationStatus,
```

```
312         CAN0_SIG_InsulationValue,
313
314         CAN0_SIG_MovAverage_Power_1s,
315         CAN0_SIG_MovAverage_Power_5s,
316         CAN0_SIG_MovAverage_Power_10s,
317         CAN0_SIG_MovAverage_Power_30s,
318         CAN0_SIG_MovAverage_Power_60s,
319         CAN0_SIG_MovAverage_Power_config,
320
321         CAN0_SIG_MovAverage_Current_1s,
322         CAN0_SIG_MovAverage_Current_5s,
323         CAN0_SIG_MovAverage_Current_10s,
324         CAN0_SIG_MovAverage_Current_30s,
325         CAN0_SIG_MovAverage_Current_60s,
326         CAN0_SIG_MovAverage_Current_config,
327
328         CAN0_SIG_PackVolt_Battery,
329         CAN0_SIG_PackVolt_PowerNet,
330
331         CAN0_SIG_Mod0_volt_valid_0_2,
332         CAN0_SIG_Mod0_volt_0,
333         CAN0_SIG_Mod0_volt_1,
334         CAN0_SIG_Mod0_volt_2,
335         CAN0_SIG_Mod0_volt_valid_3_5,
336         CAN0_SIG_Mod0_volt_3,
337         CAN0_SIG_Mod0_volt_4,
338         CAN0_SIG_Mod0_volt_5,
339         CAN0_SIG_Mod0_volt_valid_6_8,
340         CAN0_SIG_Mod0_volt_6,
341         CAN0_SIG_Mod0_volt_7,
342         CAN0_SIG_Mod0_volt_8,
343         CAN0_SIG_Mod0_volt_valid_9_11,
344         CAN0_SIG_Mod0_volt_9,
345         CAN0_SIG_Mod0_volt_10,
346         CAN0_SIG_Mod0_volt_11,
347         CAN0_SIG_Mod0_volt_valid_12_14,
348         CAN0_SIG_Mod0_volt_12,
349         CAN0_SIG_Mod0_volt_13,
350         CAN0_SIG_Mod0_volt_14,
351         CAN0_SIG_Mod0_volt_valid_15_17,
352         CAN0_SIG_Mod0_volt_15,
353         CAN0_SIG_Mod0_volt_16,
354         CAN0_SIG_Mod0_volt_17,
355
356         CAN0_SIG_Mod0_temp_valid_0_2,
357         CAN0_SIG_Mod0_temp_0,
358         CAN0_SIG_Mod0_temp_1,
359         CAN0_SIG_Mod0_temp_2,
360         CAN0_SIG_Mod0_temp_valid_3_5,
361         CAN0_SIG_Mod0_temp_3,
362         CAN0_SIG_Mod0_temp_4,
363         CAN0_SIG_Mod0_temp_5,
```

```
364         CAN0_SIG_Mod0_temp_valid_6_8,
365         CAN0_SIG_Mod0_temp_6,
366         CAN0_SIG_Mod0_temp_7,
367         CAN0_SIG_Mod0_temp_8,
368         CAN0_SIG_Mod0_temp_valid_9_11,
369         CAN0_SIG_Mod0_temp_9,
370         CAN0_SIG_Mod0_temp_10,
371         CAN0_SIG_Mod0_temp_11,
372
373         CAN0_SIG_Mod1_volt_valid_0_2,
374         CAN0_SIG_Mod1_volt_0,
375         CAN0_SIG_Mod1_volt_1,
376         CAN0_SIG_Mod1_volt_2,
377         CAN0_SIG_Mod1_volt_valid_3_5,
378         CAN0_SIG_Mod1_volt_3,
379         CAN0_SIG_Mod1_volt_4,
380         CAN0_SIG_Mod1_volt_5,
381         CAN0_SIG_Mod1_volt_valid_6_8,
382         CAN0_SIG_Mod1_volt_6,
383         CAN0_SIG_Mod1_volt_7,
384         CAN0_SIG_Mod1_volt_8,
385         CAN0_SIG_Mod1_volt_valid_9_11,
386         CAN0_SIG_Mod1_volt_9,
387         CAN0_SIG_Mod1_volt_10,
388         CAN0_SIG_Mod1_volt_11,
389         CAN0_SIG_Mod1_volt_valid_12_14,
390         CAN0_SIG_Mod1_volt_12,
391         CAN0_SIG_Mod1_volt_13,
392         CAN0_SIG_Mod1_volt_14,
393         CAN0_SIG_Mod1_volt_valid_15_17,
394         CAN0_SIG_Mod1_volt_15,
395         CAN0_SIG_Mod1_volt_16,
396         CAN0_SIG_Mod1_volt_17,
397
398         CAN0_SIG_Mod1_temp_valid_0_2,
399         CAN0_SIG_Mod1_temp_0,
400         CAN0_SIG_Mod1_temp_1,
401         CAN0_SIG_Mod1_temp_2,
402         CAN0_SIG_Mod1_temp_valid_3_5,
403         CAN0_SIG_Mod1_temp_3,
404         CAN0_SIG_Mod1_temp_4,
405         CAN0_SIG_Mod1_temp_5,
406         CAN0_SIG_Mod1_temp_valid_6_8,
407         CAN0_SIG_Mod1_temp_6,
408         CAN0_SIG_Mod1_temp_7,
409         CAN0_SIG_Mod1_temp_8,
410         CAN0_SIG_Mod1_temp_valid_9_11,
411         CAN0_SIG_Mod1_temp_9,
412         CAN0_SIG_Mod1_temp_10,
413         CAN0_SIG_Mod1_temp_11,
414
415         CAN0_SIG_Mod2_volt_valid_0_2,
```

```
416       CAN0_SIG_Mod2_volt_0,
417       CAN0_SIG_Mod2_volt_1,
418       CAN0_SIG_Mod2_volt_2,
419       CAN0_SIG_Mod2_volt_valid_3_5,
420       CAN0_SIG_Mod2_volt_3,
421       CAN0_SIG_Mod2_volt_4,
422       CAN0_SIG_Mod2_volt_5,
423       CAN0_SIG_Mod2_volt_valid_6_8,
424       CAN0_SIG_Mod2_volt_6,
425       CAN0_SIG_Mod2_volt_7,
426       CAN0_SIG_Mod2_volt_8,
427       CAN0_SIG_Mod2_volt_valid_9_11,
428       CAN0_SIG_Mod2_volt_9,
429       CAN0_SIG_Mod2_volt_10,
430       CAN0_SIG_Mod2_volt_11,
431       CAN0_SIG_Mod2_volt_valid_12_14,
432       CAN0_SIG_Mod2_volt_12,
433       CAN0_SIG_Mod2_volt_13,
434       CAN0_SIG_Mod2_volt_14,
435       CAN0_SIG_Mod2_volt_valid_15_17,
436       CAN0_SIG_Mod2_volt_15,
437       CAN0_SIG_Mod2_volt_16,
438       CAN0_SIG_Mod2_volt_17,
439
440       CAN0_SIG_Mod2_temp_valid_0_2,
441       CAN0_SIG_Mod2_temp_0,
442       CAN0_SIG_Mod2_temp_1,
443       CAN0_SIG_Mod2_temp_2,
444       CAN0_SIG_Mod2_temp_valid_3_5,
445       CAN0_SIG_Mod2_temp_3,
446       CAN0_SIG_Mod2_temp_4,
447       CAN0_SIG_Mod2_temp_5,
448       CAN0_SIG_Mod2_temp_valid_6_8,
449       CAN0_SIG_Mod2_temp_6,
450       CAN0_SIG_Mod2_temp_7,
451       CAN0_SIG_Mod2_temp_8,
452       CAN0_SIG_Mod2_temp_valid_9_11,
453       CAN0_SIG_Mod2_temp_9,
454       CAN0_SIG_Mod2_temp_10,
455       CAN0_SIG_Mod2_temp_11,
456
457       CAN0_SIG_Mod3_volt_valid_0_2,
458       CAN0_SIG_Mod3_volt_0,
459       CAN0_SIG_Mod3_volt_1,
460       CAN0_SIG_Mod3_volt_2,
461       CAN0_SIG_Mod3_volt_valid_3_5,
462       CAN0_SIG_Mod3_volt_3,
463       CAN0_SIG_Mod3_volt_4,
464       CAN0_SIG_Mod3_volt_5,
465       CAN0_SIG_Mod3_volt_valid_6_8,
466       CAN0_SIG_Mod3_volt_6,
467       CAN0_SIG_Mod3_volt_7,
```

```
468       CAN0_SIG_Mod3_volt_8,
469       CAN0_SIG_Mod3_volt_valid_9_11,
470       CAN0_SIG_Mod3_volt_9,
471       CAN0_SIG_Mod3_volt_10,
472       CAN0_SIG_Mod3_volt_11,
473       CAN0_SIG_Mod3_volt_valid_12_14,
474       CAN0_SIG_Mod3_volt_12,
475       CAN0_SIG_Mod3_volt_13,
476       CAN0_SIG_Mod3_volt_14,
477       CAN0_SIG_Mod3_volt_valid_15_17,
478       CAN0_SIG_Mod3_volt_15,
479       CAN0_SIG_Mod3_volt_16,
480       CAN0_SIG_Mod3_volt_17,
481
482       CAN0_SIG_Mod3_temp_valid_0_2,
483       CAN0_SIG_Mod3_temp_0,
484       CAN0_SIG_Mod3_temp_1,
485       CAN0_SIG_Mod3_temp_2,
486       CAN0_SIG_Mod3_temp_valid_3_5,
487       CAN0_SIG_Mod3_temp_3,
488       CAN0_SIG_Mod3_temp_4,
489       CAN0_SIG_Mod3_temp_5,
490       CAN0_SIG_Mod3_temp_valid_6_8,
491       CAN0_SIG_Mod3_temp_6,
492       CAN0_SIG_Mod3_temp_7,
493       CAN0_SIG_Mod3_temp_8,
494       CAN0_SIG_Mod3_temp_valid_9_11,
495       CAN0_SIG_Mod3_temp_9,
496       CAN0_SIG_Mod3_temp_10,
497       CAN0_SIG_Mod3_temp_11,
498
499       CAN0_SIG_Mod4_volt_valid_0_2,
500       CAN0_SIG_Mod4_volt_0,
501       CAN0_SIG_Mod4_volt_1,
502       CAN0_SIG_Mod4_volt_2,
503       CAN0_SIG_Mod4_volt_valid_3_5,
504       CAN0_SIG_Mod4_volt_3,
505       CAN0_SIG_Mod4_volt_4,
506       CAN0_SIG_Mod4_volt_5,
507       CAN0_SIG_Mod4_volt_valid_6_8,
508       CAN0_SIG_Mod4_volt_6,
509       CAN0_SIG_Mod4_volt_7,
510       CAN0_SIG_Mod4_volt_8,
511       CAN0_SIG_Mod4_volt_valid_9_11,
512       CAN0_SIG_Mod4_volt_9,
513       CAN0_SIG_Mod4_volt_10,
514       CAN0_SIG_Mod4_volt_11,
515       CAN0_SIG_Mod4_volt_valid_12_14,
516       CAN0_SIG_Mod4_volt_12,
517       CAN0_SIG_Mod4_volt_13,
518       CAN0_SIG_Mod4_volt_14,
519       CAN0_SIG_Mod4_volt_valid_15_17,
```

```
520       CAN0_SIG_Mod4_volt_15,
521       CAN0_SIG_Mod4_volt_16,
522       CAN0_SIG_Mod4_volt_17,
523
524       CAN0_SIG_Mod4_temp_valid_0_2,
525       CAN0_SIG_Mod4_temp_0,
526       CAN0_SIG_Mod4_temp_1,
527       CAN0_SIG_Mod4_temp_2,
528       CAN0_SIG_Mod4_temp_valid_3_5,
529       CAN0_SIG_Mod4_temp_3,
530       CAN0_SIG_Mod4_temp_4,
531       CAN0_SIG_Mod4_temp_5,
532       CAN0_SIG_Mod4_temp_valid_6_8,
533       CAN0_SIG_Mod4_temp_6,
534       CAN0_SIG_Mod4_temp_7,
535       CAN0_SIG_Mod4_temp_8,
536       CAN0_SIG_Mod4_temp_valid_9_11,
537       CAN0_SIG_Mod4_temp_9,
538       CAN0_SIG_Mod4_temp_10,
539       CAN0_SIG_Mod4_temp_11,
540
541       CAN0_SIG_Mod5_volt_valid_0_2,
542       CAN0_SIG_Mod5_volt_0,
543       CAN0_SIG_Mod5_volt_1,
544       CAN0_SIG_Mod5_volt_2,
545       CAN0_SIG_Mod5_volt_valid_3_5,
546       CAN0_SIG_Mod5_volt_3,
547       CAN0_SIG_Mod5_volt_4,
548       CAN0_SIG_Mod5_volt_5,
549       CAN0_SIG_Mod5_volt_valid_6_8,
550       CAN0_SIG_Mod5_volt_6,
551       CAN0_SIG_Mod5_volt_7,
552       CAN0_SIG_Mod5_volt_8,
553       CAN0_SIG_Mod5_volt_valid_9_11,
554       CAN0_SIG_Mod5_volt_9,
555       CAN0_SIG_Mod5_volt_10,
556       CAN0_SIG_Mod5_volt_11,
557       CAN0_SIG_Mod5_volt_valid_12_14,
558       CAN0_SIG_Mod5_volt_12,
559       CAN0_SIG_Mod5_volt_13,
560       CAN0_SIG_Mod5_volt_14,
561       CAN0_SIG_Mod5_volt_valid_15_17,
562       CAN0_SIG_Mod5_volt_15,
563       CAN0_SIG_Mod5_volt_16,
564       CAN0_SIG_Mod5_volt_17,
565
566       CAN0_SIG_Mod5_temp_valid_0_2,
567       CAN0_SIG_Mod5_temp_0,
568       CAN0_SIG_Mod5_temp_1,
569       CAN0_SIG_Mod5_temp_2,
570       CAN0_SIG_Mod5_temp_valid_3_5,
571       CAN0_SIG_Mod5_temp_3,
```

```
572       CAN0_SIG_Mod5_temp_4,
573       CAN0_SIG_Mod5_temp_5,
574       CAN0_SIG_Mod5_temp_valid_6_8,
575       CAN0_SIG_Mod5_temp_6,
576       CAN0_SIG_Mod5_temp_7,
577       CAN0_SIG_Mod5_temp_8,
578       CAN0_SIG_Mod5_temp_valid_9_11,
579       CAN0_SIG_Mod5_temp_9,
580       CAN0_SIG_Mod5_temp_10,
581       CAN0_SIG_Mod5_temp_11,
582
583       CAN0_SIG_Mod6_volt_valid_0_2,
584       CAN0_SIG_Mod6_volt_0,
585       CAN0_SIG_Mod6_volt_1,
586       CAN0_SIG_Mod6_volt_2,
587       CAN0_SIG_Mod6_volt_valid_3_5,
588       CAN0_SIG_Mod6_volt_3,
589       CAN0_SIG_Mod6_volt_4,
590       CAN0_SIG_Mod6_volt_5,
591       CAN0_SIG_Mod6_volt_valid_6_8,
592       CAN0_SIG_Mod6_volt_6,
593       CAN0_SIG_Mod6_volt_7,
594       CAN0_SIG_Mod6_volt_8,
595       CAN0_SIG_Mod6_volt_valid_9_11,
596       CAN0_SIG_Mod6_volt_9,
597       CAN0_SIG_Mod6_volt_10,
598       CAN0_SIG_Mod6_volt_11,
599       CAN0_SIG_Mod6_volt_valid_12_14,
600       CAN0_SIG_Mod6_volt_12,
601       CAN0_SIG_Mod6_volt_13,
602       CAN0_SIG_Mod6_volt_14,
603       CAN0_SIG_Mod6_volt_valid_15_17,
604       CAN0_SIG_Mod6_volt_15,
605       CAN0_SIG_Mod6_volt_16,
606       CAN0_SIG_Mod6_volt_17,
607
608       CAN0_SIG_Mod6_temp_valid_0_2,
609       CAN0_SIG_Mod6_temp_0,
610       CAN0_SIG_Mod6_temp_1,
611       CAN0_SIG_Mod6_temp_2,
612       CAN0_SIG_Mod6_temp_valid_3_5,
613       CAN0_SIG_Mod6_temp_3,
614       CAN0_SIG_Mod6_temp_4,
615       CAN0_SIG_Mod6_temp_5,
616       CAN0_SIG_Mod6_temp_valid_6_8,
617       CAN0_SIG_Mod6_temp_6,
618       CAN0_SIG_Mod6_temp_7,
619       CAN0_SIG_Mod6_temp_8,
620       CAN0_SIG_Mod6_temp_valid_9_11,
621       CAN0_SIG_Mod6_temp_9,
622       CAN0_SIG_Mod6_temp_10,
623       CAN0_SIG_Mod6_temp_11,
```

```
624
625         CAN0_SIG_Mod7_volt_valid_0_2,
626         CAN0_SIG_Mod7_volt_0,
627         CAN0_SIG_Mod7_volt_1,
628         CAN0_SIG_Mod7_volt_2,
629         CAN0_SIG_Mod7_volt_valid_3_5,
630         CAN0_SIG_Mod7_volt_3,
631         CAN0_SIG_Mod7_volt_4,
632         CAN0_SIG_Mod7_volt_5,
633         CAN0_SIG_Mod7_volt_valid_6_8,
634         CAN0_SIG_Mod7_volt_6,
635         CAN0_SIG_Mod7_volt_7,
636         CAN0_SIG_Mod7_volt_8,
637         CAN0_SIG_Mod7_volt_valid_9_11,
638         CAN0_SIG_Mod7_volt_9,
639         CAN0_SIG_Mod7_volt_10,
640         CAN0_SIG_Mod7_volt_11,
641         CAN0_SIG_Mod7_volt_valid_12_14,
642         CAN0_SIG_Mod7_volt_12,
643         CAN0_SIG_Mod7_volt_13,
644         CAN0_SIG_Mod7_volt_14,
645         CAN0_SIG_Mod7_volt_valid_15_17,
646         CAN0_SIG_Mod7_volt_15,
647         CAN0_SIG_Mod7_volt_16,
648         CAN0_SIG_Mod7_volt_17,
649
650         CAN0_SIG_Mod7_temp_valid_0_2,
651         CAN0_SIG_Mod7_temp_0,
652         CAN0_SIG_Mod7_temp_1,
653         CAN0_SIG_Mod7_temp_2,
654         CAN0_SIG_Mod7_temp_valid_3_5,
655         CAN0_SIG_Mod7_temp_3,
656         CAN0_SIG_Mod7_temp_4,
657         CAN0_SIG_Mod7_temp_5,
658         CAN0_SIG_Mod7_temp_valid_6_8,
659         CAN0_SIG_Mod7_temp_6,
660         CAN0_SIG_Mod7_temp_7,
661         CAN0_SIG_Mod7_temp_8,
662         CAN0_SIG_Mod7_temp_valid_9_11,
663         CAN0_SIG_Mod7_temp_9,
664         CAN0_SIG_Mod7_temp_10,
665         CAN0_SIG_Mod7_temp_11,
666
667 #ifdef CURRENT_SENSOR_ISABELLENHUETTE_TRIGGERED
668         CAN0_SIG_ISA_Trigger,
669 #endif /* CURRENT_SENSOR_ISABELLENHUETTE_TRIGGERED */
670
671         CAN0_SIGNAL_NONE = 0xFFFF
672 } CANS_CAN0_signalsTx_e;
673
674 /**
675  * symbolic names for CAN1 transmission signals
```

```
 */
typedef enum {
    CAN1_TX_SIGNAL_NONE = 0xFFFF,
} CANS_CAN1_signalsTx_e;


/**
 * symbolic names for CAN 0 receive signals
 */
typedef enum {
    CAN0_SIG_ReceiveStateRequest,          /*!< current sensor counter */
    CAN0_SIG_IVT_Current_MuxID,            /*!< current sensor measurement type */
    CAN0_SIG_IVT_Current_Status,           /*!< current sensor counter */
    CAN0_SIG_IVT_Current_Measurement,      /*!< current sensor measurement I */
    CAN0_SIG_IVT_Voltage_1_MuxID,          /*!< current sensor measurement type */
    CAN0_SIG_IVT_Voltage_1_Status,         /*!< current sensor counter */
    CAN0_SIG_IVT_Voltage_1_Measurement,    /*!< current sensor measurement U1 */
    CAN0_SIG_IVT_Voltage_2_MuxID,          /*!< current sensor measurement type */
    CAN0_SIG_IVT_Voltage_2_Status,         /*!< current sensor counter */
    CAN0_SIG_IVT_Voltage_2_Measurement,    /*!< current sensor measurement U2 */
    CAN0_SIG_IVT_Voltage_3_MuxID,          /*!< current sensor measurement type */
    CAN0_SIG_IVT_Voltage_3_Status,         /*!< current sensor counter */
    CAN0_SIG_IVT_Voltage_3_Measurement,    /*!< current sensor measurement U3 */
    CAN0_SIG_IVT_Temperature_MuxID,        /*!< current sensor measurement type */
    CAN0_SIG_IVT_Temperature_Status,       /*!< current sensor counter */
    CAN0_SIG_IVT_Temperature_Measurement,  /*!< current sensor measurement T */
    CAN0_SIG_IVT_Power_MuxID,              /*!< current sensor measurement type */
    CAN0_SIG_IVT_Power_Status,             /*!< current sensor counter */
    CAN0_SIG_IVT_Power_Measurement,        /*!< current sensor measurement P */
    CAN0_SIG_IVT_CC_MuxID,                 /*!< current sensor measurement type */
    CAN0_SIG_IVT_CC_Status,                /*!< current sensor counter */
    CAN0_SIG_IVT_CC_Measurement,           /*!< current sensor measurement C-C */
    CAN0_SIG_IVT_EC_MuxID,                 /*!< current sensor measurement type */
    CAN0_SIG_IVT_EC_Status,                /*!< current sensor counter */
    CAN0_SIG_IVT_EC_Measurement,           /*!< current sensor measurement E-C */
    CAN0_SIG_DEBUG_Data,                   /*!< Data of debug message */
    CAN0_SIG_GetReleaseVersion,
    CAN0_SIG_ReceiveEngineRequest,
} CANS_CAN0_signalsRx_e;


/**
 * symbolic names for CAN 1 receive signals
 */
typedef enum {
    CAN1_RX_SIGNAL_NONE = 0xFFFF,
} CANS_CAN1_signalsRx_e;

typedef enum {
    CAN_RX_DIRECTION = 0,
    CAN_TX_DIRECTION = 1
} CANS_messageDirection_t;
```

```c

typedef enum {
    littleEndian = 0,
    bigEndian = 1
} CANS_byteOrder_e;

typedef union {
    CANS_messagesTx_e Tx;
    CANS_messagesRx_e Rx;
} CANS_messages_t;

typedef union {
    CANS_CAN0_signalsTx_e Tx;
    CANS_CAN0_signalsRx_e Rx;
} CANS_signals_t;

/**
 * type definition for structure of a CAN signal
 *
 * until now, multiplexed signal handling is hard coded
 * in the corresponding getters/setters. For use of multiplexed
 * signals refer to description in documentation.
 *
 * support for automatic scaling is planned, but not implemented yet,
 * so min, max, factor and offset are not relevant.
 */
typedef struct  {
    CANS_messages_t msgIdx;
    uint8_t bit_position;
    uint8_t bit_length;
    float min;
    float max;
    float factor;
    float offset;
    CANS_byteOrder_e byteOrder;
    can_callback_funcPtr callback;
} CANS_signal_s;

/*================== Constant and Variable Definitions ====================*/

/**
 * array for transmission CAN0 signals definition
 */
extern const CANS_signal_s cans_CAN0_signals_tx[];

/**
 * array for transmission CAN1 signals definition
 */
extern const CANS_signal_s cans_CAN1_signals_tx[];

/**
 * array for received CAN0 signals definition
```

CANS_signals_t is a union of "integers", and CANS_signal_s is a struct of true CAN signals.

cansignal_cfg.h | can_cfg.c | can.h | can.c | can_cfg.h

mcu-primary > src > driver > config > can_cfg.h > can_callback_funcPtr

```c
307    typedef uint32_t (*can_callback_funcPtr)(uint32_t idx, void * value);
```

```
 */
extern const CANS_signal_s cans_CAN0_signals_rx[];

/**
 * array for received CAN1 signals definition
 */
extern const CANS_signal_s cans_CAN1_signals_rx[];

/**
 * length of the array for the CAN0 tx signals
 */
extern const uint16_t cans_CAN0_signals_tx_length;

/**
 * length of the array for the CAN1 tx signals
 */
extern const uint16_t cans_CAN1_signals_tx_length;

/**
 * length of the array for the CAN0 rx signals
 */
extern const uint16_t cans_CAN0_signals_rx_length;

/**
 * length of the array for the CAN1 rx signals
 */
extern const uint16_t cans_CAN1_signals_rx_length;

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

 NO function prototypes. Calling of the functions are performed via using the function pointers to call back functions.
/*================== Function Implementations ============================*/

#endif /* CANSIGNAL_CFG_H_ */

```