```c
/**
 * Copyright (c) 2018 - 2019, Nordic Semiconductor ASA
 *
 * All rights reserved.
 *
 * 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, except as embedded into a Nordic
 *    Semiconductor ASA integrated circuit in a product or a software update for
 *    such product, 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 Nordic Semiconductor ASA nor the names of its
 *    contributors may be used to endorse or promote products derived from this
 *    software without specific prior written permission.
 *
 * 4. This software, with or without modification, must only be used with a
 *    Nordic Semiconductor ASA integrated circuit.
 *
 * 5. Any software provided in binary form under this license must not be reverse
 *    engineered, decompiled, modified and/or disassembled.
 *
 * THIS SOFTWARE IS PROVIDED BY NORDIC SEMICONDUCTOR ASA "AS IS" AND ANY EXPRESS
 * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL NORDIC SEMICONDUCTOR ASA 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.
 *
 */
#ifndef ZB_MULTI_SENSOR_H__
#define ZB_MULTI_SENSOR_H__

#include "zboss_api.h"
#include "zboss_api_addons.h"
#include "zb_zcl_pressure_measurement.h"

#ifdef __cplusplus
extern "C" {
#endif

#define USER_LED LED_3 //13

/* Basic cluster attributes initial values. For more information, see section 3.2.2.2 of the ZCL specification. */
#define SENSOR_INIT_BASIC_APP_VERSION       01                                  /**< Version of the application software (1 byte). */
#define SENSOR_INIT_BASIC_STACK_VERSION     10                                  /**< Version of the implementation of the Zigbee stack
#define SENSOR_INIT_BASIC_HW_VERSION        11                                  /**< Version of the hardware of the device (1 byte). */
#define SENSOR_INIT_BASIC_MANUF_NAME        "Nordic"                            /**< Manufacturer name (32 bytes). */
#define SENSOR_INIT_BASIC_MODEL_ID          "NRF_MultiSensor_bind"              /**< Model number assigned by the manufacturer (32-byte
#define SENSOR_INIT_BASIC_DATE_CODE         "20180921"                          /**< Date provided by the manufacturer of the device in
#define SENSOR_INIT_BASIC_POWER_SOURCE      ZB_ZCL_BASIC_POWER_SOURCE_DC_SOURCE /**< Type of power source or sources available for the
#define SENSOR_INIT_BASIC_LOCATION_DESC     "Office desk"                       /**< Description of the physical location of the device
#define SENSOR_INIT_BASIC_PH_ENV            ZB_ZCL_BASIC_ENV_UNSPECIFIED        /**< Description of the type of physical environment. F

#define MULTI_SENSOR_ENDPOINT               10                                  /**< Device endpoint. Used to receive light controlling

/* Main application customizable context. Stores all settings and static values. */
typedef struct
{
    zb_zcl_basic_attrs_ext_t            basic_attr;
    zb_zcl_identify_attrs_t             identify_attr;
    zb_zcl_temp_measurement_attrs_t     temp_attr;
    zb_zcl_pressure_measurement_attrs_t pres_attr;
} sensor_device_ctx_t;

#define ZB_MULTI_SENSOR_REPORT_ATTR_COUNT  2                                    /**< Number of attributes mandatory for reporting in th
#define ZB_DEVICE_VER_MULTI_SENSOR         0                                    /**< Multisensor device version. */
#define ZB_MULTI_SENSOR_IN_CLUSTER_NUM     4                                    /**< Number of the input (server) clusters in the multi
#define ZB_MULTI_SENSOR_OUT_CLUSTER_NUM    1                                    /**< Number of the output (client) clusters in the mult

/** @brief Declares cluster list for the multisensor device.
 *
 *  @param cluster_list_name            Cluster list variable name.
 *  @param pressure_measure_attr_list   Attribute list for the Pressure Measurement cluster.
 */
#define ZB_DECLARE_MULTI_SENSOR_CLUSTER_LIST(                       \
      cluster_list_name,                                            \
      basic_attr_list,                                              \
      identify_attr_list,                                           \
      temp_measure_attr_list,                                       \
      pres_measure_attr_list)                                       \
      zb_zcl_cluster_desc_t cluster_list_name[] =                   \
      {                                                             \
        ZB_ZCL_CLUSTER_DESC(                                        \
          ZB_ZCL_CLUSTER_ID_IDENTIFY,                               \
          ZB_ZCL_ARRAY_SIZE(identify_attr_list, zb_zcl_attr_t),     \
          (identify_attr_list),                                     \
          ZB_ZCL_CLUSTER_SERVER_ROLE,                               \
          ZB_ZCL_MANUF_CODE_INVALID                                 \
        ),                                                          \
        ZB_ZCL_CLUSTER_DESC(                                        \
          ZB_ZCL_CLUSTER_ID_BASIC,                                  \
          ZB_ZCL_ARRAY_SIZE(basic_attr_list, zb_zcl_attr_t),        \
          (basic_attr_list),                                        \
          ZB_ZCL_CLUSTER_SERVER_ROLE,                               \
          ZB_ZCL_MANUF_CODE_INVALID                                 \
        ),                                                          \
        ZB_ZCL_CLUSTER_DESC(                                        \
          ZB_ZCL_CLUSTER_ID_TEMP_MEASUREMENT,                       \
          ZB_ZCL_ARRAY_SIZE(temp_measure_attr_list, zb_zcl_attr_t), \
          (temp_measure_attr_list),                                 \
          ZB_ZCL_CLUSTER_SERVER_ROLE,                               \
          ZB_ZCL_MANUF_CODE_INVALID                                 \
        ),                                                          \
        ZB_ZCL_CLUSTER_DESC(                                        \
          ZB_ZCL_CLUSTER_ID_PRESSURE_MEASUREMENT,                   \
          ZB_ZCL_ARRAY_SIZE(pres_measure_attr_list, zb_zcl_attr_t), \
          (pres_measure_attr_list),                                 \
          ZB_ZCL_CLUSTER_SERVER_ROLE,                               \
          ZB_ZCL_MANUF_CODE_INVALID                                 \
        ),                                                          \
        ZB_ZCL_CLUSTER_DESC(                                        \
          ZB_ZCL_CLUSTER_ID_IDENTIFY,                               \
          0,                                                        \
          NULL,                                                     \
          ZB_ZCL_CLUSTER_CLIENT_ROLE,                               \
          ZB_ZCL_MANUF_CODE_INVALID                                 \
        )                                                           \
      }

/** @brief Declares simple descriptor for the "Device_name" device.
 *
 *  @param ep_name          Endpoint variable name.
 *  @param ep_id            Endpoint ID.
 *  @param in_clust_num     Number of the supported input clusters.
 *  @param out_clust_num    Number of the supported output clusters.
 */
#define ZB_ZCL_DECLARE_MULTI_SENSOR_DESC(ep_name, ep_id, in_clust_num, out_clust_num) \
  ZB_DECLARE_SIMPLE_DESC(in_clust_num, out_clust_num);                                 \
  ZB_AF_SIMPLE_DESC_TYPE(in_clust_num, out_clust_num) simple_desc_##ep_name =          \
  {                                                                                    \
    ep_id,                                                                             \
    ZB_AF_HA_PROFILE_ID,                                                               \
    ZB_HA_TEMPERATURE_SENSOR_DEVICE_ID,                                                \
    ZB_DEVICE_VER_MULTI_SENSOR,                                                        \
    0,                                                                                 \
    in_clust_num,                                                                      \
    out_clust_num,                                                                     \
    {                                                                                  \
      ZB_ZCL_CLUSTER_ID_BASIC,                                                         \
      ZB_ZCL_CLUSTER_ID_IDENTIFY,                                                      \
      ZB_ZCL_CLUSTER_ID_TEMP_MEASUREMENT,                                              \
      ZB_ZCL_CLUSTER_ID_PRESSURE_MEASUREMENT,                                          \
      ZB_ZCL_CLUSTER_ID_IDENTIFY,                                                      \
    }                                                                                  \
  }

/** @brief Declares endpoint for the multisensor device.
 *
 *  @param ep_name          Endpoint variable name.
 *  @param ep_id            Endpoint ID.
 *  @param cluster_list     Endpoint cluster list.
 */
#define ZB_ZCL_DECLARE_MULTI_SENSOR_EP(ep_name, ep_id, cluster_list)               \
  ZB_ZCL_DECLARE_MULTI_SENSOR_DESC(ep_name,                                        \
  ZBOSS_DEVICE_DECLARE_REPORTING_CTX(reporting_info## device_ctx_name,            \
                                    ZB_MULTI_SENSOR_REPORT_ATTR_COUNT);           \
  ZB_AF_DECLARE_ENDPOINT_DESC(ep_name, ep_id,                                     \
      ZB_AF_HA_PROFILE_ID,                                                        \
      0,                                                                          \
      NULL,                                                                       \
      ZB_ZCL_ARRAY_SIZE(cluster_list, zb_zcl_cluster_desc_t),                     \
      cluster_list,                                                               \
      (zb_af_simple_desc_1_1_t*)&simple_desc_##ep_name,                           \
      ZB_MULTI_SENSOR_REPORT_ATTR_COUNT, reporting_info## device_ctx_name, 0, NULL)


#ifdef __cplusplus
}
#endif
#endif // ZB_MULTI_SENSOR_H__
```