```
 *
 * @param[in] dev   :Structure instance of bme680_dev.
 *
 * @return Result of API execution status
 * @retval zero -> Success / +ve value -> Warning / -ve value -> Error
 */
static int8_t null_ptr_check(const struct bme680_dev *dev);

/*!
 * @brief This internal API is used to check the boundary
 * conditions.
 *
 * @param[in] value :pointer to the value.
 * @param[in] min   :minimum value.
 * @param[in] max   :maximum value.
 * @param[in] dev   :Structure instance of bme680_dev.
 *
 * @return Result of API execution status
 * @retval zero -> Success / +ve value -> Warning / -ve value -> Error
 */
static int8_t boundary_check(uint8_t *value, uint8_t min, uint8_t max, struct bme680_dev *dev);

/****************** Global Function Definitions *******************************/
/*!
 *@brief This API is the entry point.
 *It reads the chip-id and calibration data from the sensor.
 */
int8_t bme680_init(struct bme680_dev *dev)
{
    int8_t rslt;

    /* Check for null pointer in the device structure*/
    rslt = null_ptr_check(dev);
    if (rslt == BME680_OK) {
        /* Soft reset to restore it to default values*/
        rslt = bme680_soft_reset(dev);
        if (rslt == BME680_OK) {
```