```
 * @brief This internal API is used to set the memory page
 * based on register address.
 *
 * The value of memory page
 *  value   | Description
 * --------|--------------
 *   0     | BME680_PAGE0_SPI
 *   1     | BME680_PAGE1_SPI
 *
 * @param[in] dev   :Structure instance of bme680_dev.
 * @param[in] reg_addr  :Contains the register address array.
 *
 * @return Result of API execution status
 * @retval zero -> Success / +ve value -> Warning / -ve value -> Error
 */
static int8_t set_mem_page(uint8_t reg_addr, struct bme680_dev *dev);

/*!
 * @brief This internal API is used to get the memory page based
 * on register address.
 *
 * The value of memory page
 *  value   | Description
 * --------|--------------
 *   0     | BME680_PAGE0_SPI
 *   1     | BME680_PAGE1_SPI
 *
 * @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 get_mem_page(struct bme680_dev *dev);

/*!
 * @brief This internal API is used to validate the device pointer for
 * null conditions.
```