```
#define BME680_GET_BITS(reg_data, bitname)  ((reg_data & (bitname##_MSK)) >> \
    (bitname##_POS))

/** Macro variant to handle the bitname position if it is zero */
#define BME680_SET_BITS_POS_0(reg_data, bitname, data) \
                ((reg_data & ~(bitname##_MSK)) | \
                (data & bitname##_MSK))
#define BME680_GET_BITS_POS_0(reg_data, bitname)  (reg_data & (bitname##_MSK))

/** Type definitions */
/*!
 * Generic communication function pointer
 * @param[in] dev_id: Place holder to store the id of the device structure
 *                    Can be used to store the index of the Chip select or
 *                    I2C address of the device.
 * @param[in] reg_addr: Used to select the register the where data needs to
 *                       be read from or written to.
 * @param[in/out] reg_data: Data array to read/write
 * @param[in] len: Length of the data array
 */
typedef int8_t (*bme680_com_fptr_t)(uint8_t dev_id, uint8_t reg_addr, uint8_t *data, uint16_t len);

/*!
 * Delay function pointer
 * @param[in] period: Time period in milliseconds
 */
typedef void (*bme680_delay_fptr_t)(uint32_t period);

/*!
 * @brief Interface selection Enumerations
 */
enum bme680_intf {
    /*! SPI interface */
    BME680_SPI_INTF,
    /*! I2C interface */
    BME680_I2C_INTF
};
```