

```
int8_t bme680_init(struct bme680_dev *dev);

/*!
 * @brief This API writes the given data to the register address
 * of the sensor.
 *
 * @param[in] reg_addr : Register address from where the data to be written.
 * @param[in] reg_data : Pointer to data buffer which is to be written
 * in the sensor.
 * @param[in] len : No of bytes of data to write..
 * @param[in] dev : Structure instance of bme680_dev.
 *
 * @return Result of API execution status
 * @retval zero -> Success / +ve value -> Warning / -ve value -> Error
 */
int8_t bme680_set_regs(const uint8_t *reg_addr, const uint8_t *reg_data, uint8_t len, struct bme680_dev *dev);

/*!
 * @brief This API reads the data from the given register address of the sensor.
 *
 * @param[in] reg_addr : Register address from where the data to be read
 * @param[out] reg_data : Pointer to data buffer to store the read data.
 * @param[in] len : No of bytes of data to be read.
 * @param[in] dev : Structure instance of bme680_dev.
 *
 * @return Result of API execution status
 * @retval zero -> Success / +ve value -> Warning / -ve value -> Error
 */
int8_t bme680_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint16_t len, struct bme680_dev *dev);

/*!
 * @brief This API performs the soft reset of the sensor.
 *
 * @param[in] dev : Structure instance of bme680_dev.
 *
 * @return Result of API execution status
 * @retval zero -> Success / +ve value -> Warning / -ve value -> Error.
```