```
                    BME680_FILTER);

            if (desired_settings & (BME680_OST_SEL | BME680_OSP_SEL)) {
                dev->tph_sett.os_temp = BME680_GET_BITS(data_array[BME680_REG_TEMP_INDEX], BME680_OST);
                dev->tph_sett.os_pres = BME680_GET_BITS(data_array[BME680_REG_PRES_INDEX], BME680_OSP);
            }

            if (desired_settings & BME680_OSH_SEL)
                dev->tph_sett.os_hum = BME680_GET_BITS_POS_0(data_array[BME680_REG_HUM_INDEX],
                    BME680_OSH);

            /* get the gas related settings */
            if (desired_settings & BME680_HCNTRL_SEL)
                dev->gas_sett.heatr_ctrl = BME680_GET_BITS_POS_0(data_array[BME680_REG_HCTRL_INDEX],
                    BME680_HCTRL);

            if (desired_settings & (BME680_RUN_GAS_SEL | BME680_NBCONV_SEL)) {
                dev->gas_sett.nb_conv = BME680_GET_BITS_POS_0(data_array[BME680_REG_NBCONV_INDEX],
                    BME680_NBCONV);
                dev->gas_sett.run_gas = BME680_GET_BITS(data_array[BME680_REG_RUN_GAS_INDEX],
                    BME680_RUN_GAS);
            }
        }
    } else {
        rslt = BME680_E_NULL_PTR;
    }

    return rslt;
}

/*!
 * @brief This API is used to set the power mode of the sensor.
 */
int8_t bme680_set_sensor_mode(struct bme680_dev *dev)
{
    int8_t rslt;
    uint8_t tmp_pow_mode;
```