```
        uint8_t reg_addr[2] = {0};
        uint8_t reg_data[2] = {0};

        if (dev->power_mode == BME680_FORCED_MODE) {
            reg_addr[0] = BME680_RES_HEAT0_ADDR;
            reg_data[0] = calc_heater_res(dev->gas_sett.heatr_temp, dev);
            reg_addr[1] = BME680_GAS_WAIT0_ADDR;
            reg_data[1] = calc_heater_dur(dev->gas_sett.heatr_dur);
            dev->gas_sett.nb_conv = 0;
        } else {
            rslt = BME680_W_DEFINE_PWR_MODE;
        }
        if (rslt == BME680_OK)
            rslt = bme680_set_regs(reg_addr, reg_data, 2, dev);
    }

    return rslt;
}

/*!
 * @brief This internal API is used to get the gas configuration of the sensor.
 * @note heatr_temp and heatr_dur values are currently register data
 * and not the actual values set
 */
static int8_t get_gas_config(struct bme680_dev *dev)
{
    int8_t rslt;
    /* starting address of the register array for burst read*/
    uint8_t reg_addr1 = BME680_ADDR_SENS_CONF_START;
    uint8_t reg_addr2 = BME680_ADDR_GAS_CONF_START;
    uint8_t reg_data = 0;

    /* Check for null pointer in the device structure*/
    rslt = null_ptr_check(dev);
    if (rslt == BME680_OK) {
        if (BME680_SPI_INTF == dev->intf) {
            /* Memory page switch the SPI address*/
```