

```
    rslt = null_ptr_check(dev);
    do {
        if (rslt == BME680_OK) {
            rslt = bme680_get_regs(((uint8_t) (BME680_FIELD0_ADDR)), buff, (uint16_t) BME680_FIELD_LENGTH,
                                   dev);

            data->status = buff[0] & BME680_NEW_DATA_MSK;
            data->gas_index = buff[0] & BME680_GAS_INDEX_MSK;
            data->meas_index = buff[1];

            /* read the raw data from the sensor */
            adc_pres = (uint32_t) (((uint32_t) buff[2] * 4096) | ((uint32_t) buff[3] * 16)
                                   | ((uint32_t) buff[4] / 16));
            adc_temp = (uint32_t) (((uint32_t) buff[5] * 4096) | ((uint32_t) buff[6] * 16)
                                   | ((uint32_t) buff[7] / 16));
            adc_hum = (uint16_t) (((uint32_t) buff[8] * 256) | (uint32_t) buff[9]);
            adc_gas_res = (uint16_t) ((uint32_t) buff[13] * 4 | (((uint32_t) buff[14]) / 64));
            gas_range = buff[14] & BME680_GAS_RANGE_MSK;

            data->status |= buff[14] & BME680_GASM_VALID_MSK;
            data->status |= buff[14] & BME680_HEAT_STAB_MSK;

            if (data->status & BME680_NEW_DATA_MSK) {
                data->temperature = calc_temperature(adc_temp, dev);
                data->pressure = calc_pressure(adc_pres, dev);
                data->humidity = calc_humidity(adc_hum, dev);
                data->gas_resistance = calc_gas_resistance(adc_gas_res, gas_range, dev);
                break;
            }
            /* Delay to poll the data */
            dev->delay_ms(BME680_POLL_PERIOD_MS);
        }
        tries--;
    } while (tries);

    if (!tries)
        rslt = BME680_W_NO_NEW_DATA;
```