```
    var2 = (var1) * (1.0f + lookup_k1_range[gas_range]/100.0f);
    var3 = 1.0f + (lookup_k2_range[gas_range]/100.0f);

    calc_gas_res = 1.0f / (float)(var3 * (0.000000125f) * (float)(1 << gas_range) * (((((float)gas_res_adc)
        - 512.0f)/var2) + 1.0f));

    return calc_gas_res;
}

/*!
 * @brief This internal API is used to calculate the
 * heater resistance value in float format
 */
static float calc_heater_res(uint16_t temp, const struct bme680_dev *dev)
{
    float var1 = 0;
    float var2 = 0;
    float var3 = 0;
    float var4 = 0;
    float var5 = 0;
    float res_heat = 0;

    if (temp > 400) /* Cap temperature */
        temp = 400;

    var1 = (((float)dev->calib.par_gh1 / (16.0f)) + 49.0f);
    var2 = ((((float)dev->calib.par_gh2 / (32768.0f)) * (0.0005f)) + 0.00235f);
    var3 = ((float)dev->calib.par_gh3 / (1024.0f));
    var4 = (var1 * (1.0f + (var2 * (float)temp)));
    var5 = (var4 + (var3 * (float)dev->amb_temp));
    res_heat = (uint8_t)(3.4f * ((var5 * (4 / (4 + (float)dev->calib.res_heat_range)) *
        (1/(1 + ((float) dev->calib.res_heat_val * 0.002f)))) - 25));

    return res_heat;
}

#endif
```