```c
/*!
 * @brief This internal API is used to calculate the Heat Resistance value.
 */
static uint8_t calc_heater_res(uint16_t temp, const struct bme680_dev *dev)
{
    uint8_t heatr_res;
    int32_t var1;
    int32_t var2;
    int32_t var3;
    int32_t var4;
    int32_t var5;
    int32_t heatr_res_x100;

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

    var1 = (((int32_t) dev->amb_temp * dev->calib.par_gh3) / 1000) * 256;
    var2 = (dev->calib.par_gh1 + 784) * (((((dev->calib.par_gh2 + 154009) * temp * 5) / 100) + 3276800) / 10);
    var3 = var1 + (var2 / 2);
    var4 = (var3 / (dev->calib.res_heat_range + 4));
    var5 = (131 * dev->calib.res_heat_val) + 65536;
    heatr_res_x100 = (int32_t) (((var4 / var5) - 250) * 34);
    heatr_res = (uint8_t) ((heatr_res_x100 + 50) / 100);

    return heatr_res;
}

#else


/*!
 * @brief This internal API is used to calculate the
 * temperature value in float format
 */
static float calc_temperature(uint32_t temp_adc, struct bme680_dev *dev)
{
    float var1 = 0;
```