

```
var2 = var1 * (((float)(((float) dev->calib.par_h2 / 262144.0f) * (1.0f + (((float)dev->calib.par_h4 / 16384.0f)
    * temp_comp) + (((float)dev->calib.par_h5 / 1048576.0f) * temp_comp * temp_comp)))));

var3 = (float) dev->calib.par_h6 / 16384.0f;

var4 = (float) dev->calib.par_h7 / 2097152.0f;

calc_hum = var2 + ((var3 + (var4 * temp_comp)) * var2 * var2);

if (calc_hum > 100.0f)
    calc_hum = 100.0f;
else if (calc_hum < 0.0f)
    calc_hum = 0.0f;

return calc_hum;
}

/*!
 * @brief This internal API is used to calculate the
 * gas resistance value in float format
 */
static float calc_gas_resistance(uint16_t gas_res_adc, uint8_t gas_range, const struct bme680_dev *dev)
{
    float calc_gas_res;
    float var1 = 0;
    float var2 = 0;
    float var3 = 0;

    const float lookup_k1_range[16] = {
        0.0, 0.0, 0.0, 0.0, 0.0, -1.0, 0.0, -0.8,
        0.0, 0.0, -0.2, -0.5, 0.0, -1.0, 0.0, 0.0};
    const float lookup_k2_range[16] = {
        0.0, 0.0, 0.0, 0.0, 0.1, 0.7, 0.0, -0.8,
        -0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

    var1 = (1340.0f + (5.0f * dev->calib.range_sw_err));
```