```c
        calc_hum = 100000;
    else if (calc_hum < 0)
        calc_hum = 0;

    return (uint32_t) calc_hum;
}

/*!
 * @brief This internal API is used to calculate the Gas Resistance value.
 */
static uint32_t calc_gas_resistance(uint16_t gas_res_adc, uint8_t gas_range, const struct bme680_dev *dev)
{
    int64_t var1;
    uint64_t var2;
    int64_t var3;
    uint32_t calc_gas_res;
    /**Look up table 1 for the possible gas range values */
    uint32_t lookupTable1[16] = { UINT32_C(2147483647), UINT32_C(2147483647), UINT32_C(2147483647), UINT32_C
      (2147483647),
        UINT32_C(2147483647), UINT32_C(2126008810), UINT32_C(2147483647), UINT32_C(2130303777),
        UINT32_C(2147483647), UINT32_C(2147483647), UINT32_C(2143188679), UINT32_C(2136746228),
        UINT32_C(2147483647), UINT32_C(2126008810), UINT32_C(2147483647), UINT32_C(2147483647) };
    /**Look up table 2 for the possible gas range values */
    uint32_t lookupTable2[16] = { UINT32_C(4096000000), UINT32_C(2048000000), UINT32_C(1024000000), UINT32_C(512000000),
        UINT32_C(255744255), UINT32_C(127110228), UINT32_C(64000000), UINT32_C(32258064), UINT32_C(16016016),
        UINT32_C(8000000), UINT32_C(4000000), UINT32_C(2000000), UINT32_C(1000000), UINT32_C(500000),
        UINT32_C(250000), UINT32_C(125000) };

    var1 = (int64_t) ((1340 + (5 * (int64_t) dev->calib.range_sw_err)) *
        ((int64_t) lookupTable1[gas_range])) >> 16;
    var2 = (((int64_t) ((int64_t) gas_res_adc << 15) - (int64_t) (16777216)) + var1);
    var3 = (((int64_t) lookupTable2[gas_range] * (int64_t) var1) >> 9);
    calc_gas_res = (uint32_t) ((var3 + ((int64_t) var2 >> 1)) / (int64_t) var2);

    return calc_gas_res;
}
```