

```

    pressure_comp = (int32_t)(pressure_comp) + ((var1 + var2 + var3 +
        ((int32_t)dev->calib.par_p7 << 7)) >> 4);

    return (uint32_t)pressure_comp;
}

/*!
 * @brief This internal API is used to calculate the humidity value.
 */
static uint32_t calc_humidity(uint16_t hum_adc, const struct bme680_dev *dev)
{
    int32_t var1;
    int32_t var2;
    int32_t var3;
    int32_t var4;
    int32_t var5;
    int32_t var6;
    int32_t temp_scaled;
    int32_t calc_hum;

    temp_scaled = (((int32_t) dev->calib.t_fine * 5) + 128) >> 8;
    var1 = (int32_t) (hum_adc - ((int32_t) ((int32_t) dev->calib.par_h1 * 16)))
        - (((temp_scaled * (int32_t) dev->calib.par_h3) / ((int32_t) 100)) >> 1);
    var2 = ((int32_t) dev->calib.par_h2
        * (((temp_scaled * (int32_t) dev->calib.par_h4) / ((int32_t) 100))
            + (((temp_scaled * ((temp_scaled * (int32_t) dev->calib.par_h5) / ((int32_t) 100))) >> 6)
                / ((int32_t) 100)) + (int32_t) (1 << 14)))) >> 10;
    var3 = var1 * var2;
    var4 = (int32_t) dev->calib.par_h6 << 7;
    var4 = ((var4) + ((temp_scaled * (int32_t) dev->calib.par_h7) / ((int32_t) 100))) >> 4;
    var5 = ((var3 >> 14) * (var4 >> 14)) >> 10;
    var6 = (var4 * var5) >> 1;
    calc_hum = (((var3 + var6) >> 10) * ((int32_t) 1000)) >> 12;

    if (calc_hum > 100000) /* Cap at 100%rH */

```