```c
    int8_t rslt;
    uint8_t mode;

    /* Check for null pointer in the device structure*/
    rslt = null_ptr_check(dev);
    if (rslt == BME680_OK) {
        rslt = bme680_get_regs(BME680_CONF_T_P_MODE_ADDR, &mode, 1, dev);
        /* Masking the other register bit info*/
        dev->power_mode = mode & BME680_MODE_MSK;
    }

    return rslt;
}

/*!
 * @brief This API is used to set the profile duration of the sensor.
 */
void bme680_set_profile_dur(uint16_t duration, struct bme680_dev *dev)
{
    uint32_t tph_dur; /* Calculate in us */
    uint32_t meas_cycles;
    uint8_t os_to_meas_cycles[6] = {0, 1, 2, 4, 8, 16};

    meas_cycles = os_to_meas_cycles[dev->tph_sett.os_temp];
    meas_cycles += os_to_meas_cycles[dev->tph_sett.os_pres];
    meas_cycles += os_to_meas_cycles[dev->tph_sett.os_hum];

    /* TPH measurement duration */
    tph_dur = meas_cycles * UINT32_C(1963);
    tph_dur += UINT32_C(477 * 4); /* TPH switching duration */
    tph_dur += UINT32_C(477 * 5); /* Gas measurement duration */
    tph_dur += UINT32_C(500); /* Get it to the closest whole number.*/
    tph_dur /= UINT32_C(1000); /* Convert to ms */

    tph_dur += UINT32_C(1); /* Wake up duration of 1ms */
    /* The remaining time should be used for heating */
    dev->gas_sett.heatr_dur = duration - (uint16_t) tph_dur;
```