



温湿度传感器 LCD 显示 DEMO

软件设计说明书

V1.0.0

版本号	修订说明	修订人	审批人	日期
V1.0.0	初版	Arien		2021-10-29

1. DP 点与 cluster 属性介绍

DPID	DP 含义	数据传输类型	数据类型	是否标准指令
1	温度	只上报(ro)	数值型(Value)	是
2	湿度	只上报(ro)	数值型(Value)	是
4	电池电量	只上报(ro)	数值型(Value)	是

cluster name	cluster id	client	server	attribute name	attribute id
Power Config	0x0001		√	Mains Voltage	0x0000
				Battery Voltage	0x0020
				Battery Percentage Remain	0x0021
Temperature Measurement	0x0402		√	Measured Value	0x0000
				Min Measured Value	0x0001
				Max Measured Value	0x0002
Relative Humidity	0x0405		√	Measured Value	0x0000
				Min Measured Value	0x0001
				Max Measured Value	0x0002
OTA Upgrade	0x0019		√	Upgrade Server ID	0x0000
				File Offset	0x0001
				Current File Version	0x0002
				Image Upgrade Status	0x0006
				Manufacturer ID	0x0007
				Image Type ID	0x0008
				Min Block Request Period	0x0009

1.1 DP1 温度

指令方向	CLUSTER ID	ATTR ID	VALUE
上报	0x0402	0x0000 Measured Value	0x0000-0xffff

1.2 DP2 湿度

指令方向	CLUSTER ID	ATTR ID	VALUE
上报	0x0405	0x0000 Measured Value	0x0000-0xffff

1.3 DP4 电池电量

指令方向	CLUSTER ID	ATTR ID	VALUE
上报	0x0001	0x0020 Battery Voltage 0x0021 Battery Percentage Remain	0x00-0xff

2. 硬件电路

外设		IO	有效电平状态
UART	uart0_rx	PA6	NULL
	uart0_tx	PA5	NULL
LED	led0	PC0	LOW
KEY	key0	PA0	LOW
SHT40	i2c_scl	PA3	LOW
	i2c_sda	PA4	LOW
VKL144B	i2c_scl	PB0	LOW
	i2c_sda	PD1	LOW

传感器设备为低功耗设备，因此 ZS3L 模块的 PC00、PC02、PD00、PD01 为非唤醒源不能作为唤醒引脚，不能作为外部中断的输入引脚。PB01 作为输入，功耗会增加 17.5uA。PC02 和 PD00 脚设为上拉模式，功耗会增加 6uA。因此为了确保低功耗，PB01，PC02 和 PD00 脚尽量不要作为输入引脚。



图 1 ZS3L 模组图

文档中心: <https://developer.tuya.com/cn/docs/iot/zs3l?id=K97r37j19f496>

3. 系统流程图

3.1 Demo 主流程图

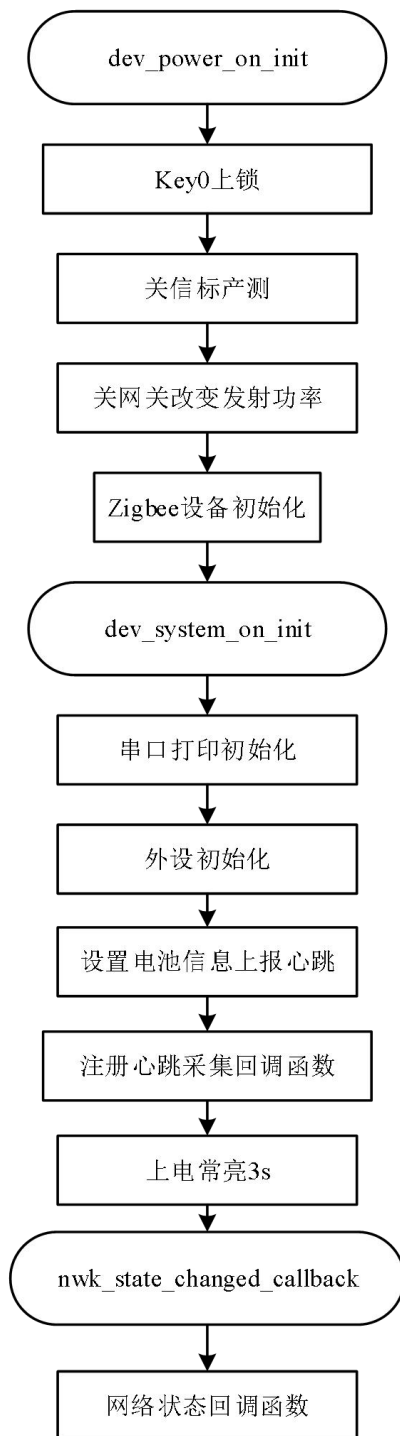


图 2 应用层主流程图

3.2 Demo 子流程图

3.2.1 设备外设初始化流程图

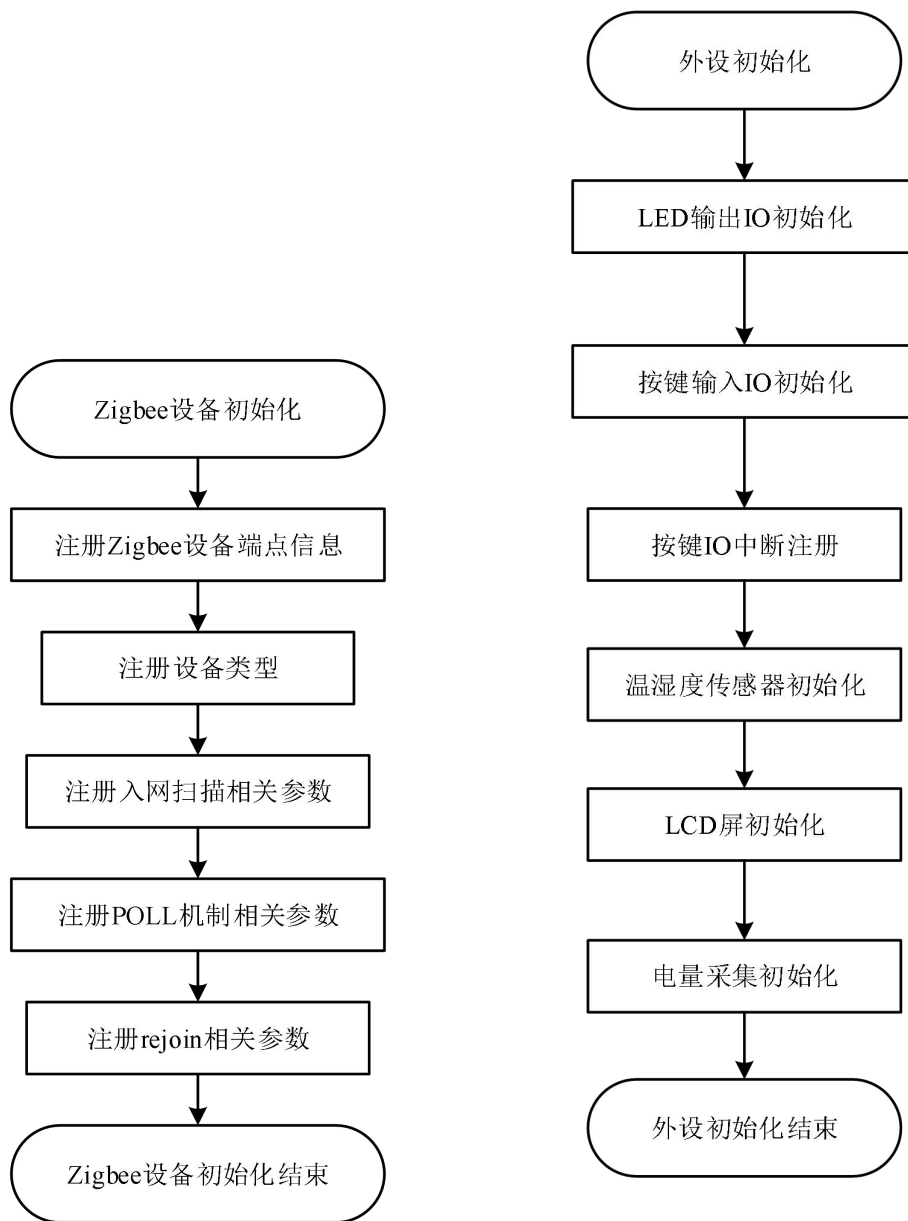


图3 外设初始化流程图

3.2.2 中断处理流程图

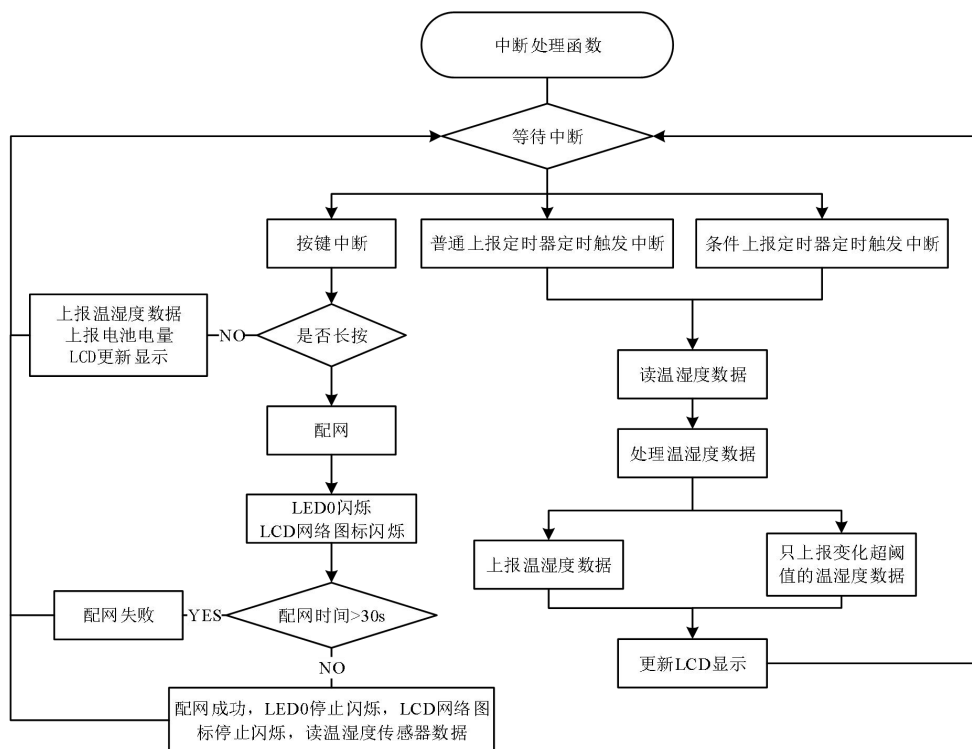


图 4 中断处理流程图

- 按键短按，触发按键中断，会先判断当前网络状态是否为 NET_JOIN_OK 或是 NET_REJOIN_OK，如果是开启定时读取上报温湿度传感器事件，并上报电池电量。如果不是，仅仅将 LED0 开启，按键释放后，LED0 关闭。
- 按键长按 3s 以上，LED0 和 LCD 的网络图标开始交替闪烁，此时松开按键，模块进入配网状态。当配网时间超过 30s 视为配网失败，LED0 关闭闪烁，同时 LCD 网络图标停止闪烁并为熄灭状态。如果配网成功，LED0 关闭闪烁，同时 LCD 网络图标停止闪烁并为亮起状态，然后开启开启定时读取上报温湿度传感器事件。
- demo 中温湿度传感器读取上报事件包含两个定时器。一个定时器达到定时时间后，定时强制上报刚采集转化后的温湿度数据，同时更新 LCD 显示。而另一个定时器达到定时时间后，会将刚采集转化后的温湿度数据与上一次温湿度数据进行对比，只有超过 temperature_humidity_sensor_init 函数中设定的变化阈值才会上报。

3.2.3 网络状态回调流程图

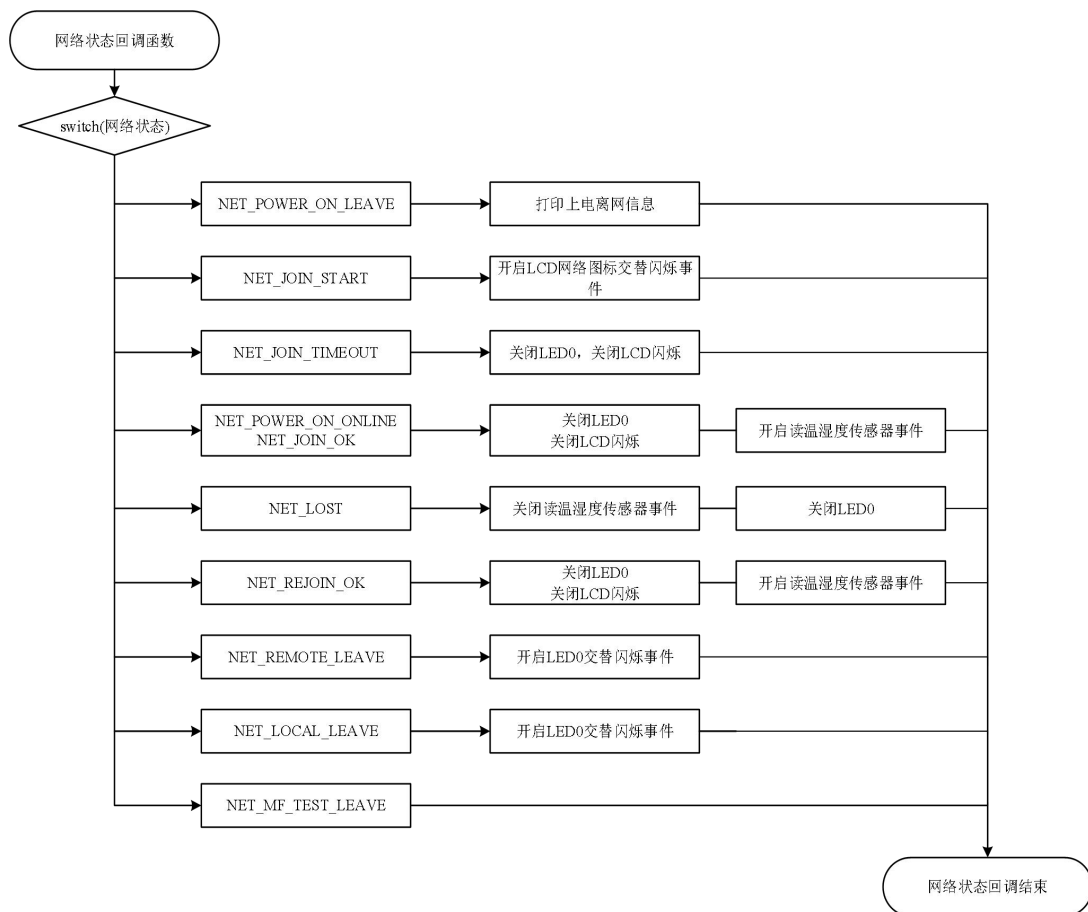


图 5 网络状态回调流程图

- 上电后先进行 Zigbee 设备初始化、IO 外设初始化、打印初始化等操作。
- 协议栈初始化后进入 system_on 之后初始化传感器，设备常亮 3s 表示上电状态。
- 电池电量初始化，按照 demo 代码可按需修改参数。
- 长按按键 3s 后松手可触发设备配网，开始配网时指示闪烁和 LCD 网络图标闪烁，配网超时后（30s）指示灯熄灭，LCD 网络图标停止闪烁。配网成功后，LCD 网络图标会停止闪烁并为常亮状态。配网成功后若再长接触发配网，设备会先本地离网后自动触发配网。
- 普通传感器上报定时器定时触发后，会读取温湿度传感器数据并做处理后进行上报，并更新 LCD 显示。条件传感器上报定时器定时触发后，则满足变化值超过设定阈值的条件才会进行上报。
- 设备 4h 会上报一次心跳，如果超过 12h 网关未收到心跳等信息会报离线。

4. 函数说明

4.1 temperature_humidity_sensor_init()

写入温湿度传感器 I2C 的 SDA 和 SCL 引脚参数

写入用到的定时器 ID

写入温湿度传感器相关配置参数

温湿度传感器芯片 SHT40 底层驱动初始化

4.2 sensor_data_report_callback()

传感器采集完数据后会调用该回调函数，进行温湿度数据上报以及 LCD 显示更新

4.3 lcd_vkl144b_init()

LCD 芯片 VKL144B 底层驱动初始化

4.4 battery_sampling_evt_callback()

根据当前电量调整合适的采样周期和电量上报心跳周期，并通过 LCD 显示当前电量。

4.5 stop_net_blink()

根据传入 bool 参数决定停止 LCD 网络图标闪烁时，LCD 网络图标的亮暗状态

4.6 ty_sht40_start_read()

I2C 对应格式读取温湿度传感器数据，并对数据进行处理上报到网关

4.7 sensor_read_according_to_cycles_tmr()

开启周期定时读温湿度传感器数据的事件，相对于 ty_sht40_start_read()函数来说，ty_sht40_start_read()只读一次数据，而此函数是周期一直读数据。

4.8 keys_evt_handler()

按键按下触发按键回调函数，同时 sdk 内部会开启定时器中断，如果保持长按状态，每 20ms 按键回调会被定时器中断打断增加 push_time 的值，然后再在定时器回调函数中调用 keys_evt_handler()。所以不松手，会多次进入按键回调。

5. 注意事项

5.1 低功耗设备特有 rejoin 参数

- a. next_rejoin_time: rejoin 的过程是分组发送 beacon，每组会发送 rejoin_try_times 个 beacon。当上一组 rejoin 失败后再次进行 rejoin 间隔的时间。
- b. wake_up_time_after_join: join 成功后，可以进行 poll 的时间间隔。
- c. wake_up_time_after_rejoin: rejoin 成功后，可以进行 poll 的时间间隔。
- d. rejoin_try_times: 每组 rejoin 中 beacon 发送的数量
- e. power_on_auto_rejoin_flag: 程序启动后是否自动进行 rejoin 的标志位
- f. auto_rejoin_send_data: 当发送数据失败后是否自动进行 rejoin 的标志位

5.2 低功耗设备特有 poll 参数

- a. poll_interval: 两次 poll 之间的间隔（poll 在抓包中表现为 Data Request 命令）
- b. wait_app_ack_time: 在这段时间包含整个 poll 和接受数据的过程。在 join, rejoin 和子设备上报数据存在三种不同的 poll 方式，因此可以通过分析三种类型以及 poll_interval 和 wait_app_ack_time 参数计算这三个过程分别的 poll 次数
- c. poll_forever_flag: 是否一直进行 poll，如果该标志位为 1，则 wait_app_ack_time 将没有意义。所以默认值为 0，才可以限定 poll 的次数
- d. poll_failed_times: 每次 poll 后，无论网关有无数据都会回应 ack，一旦网关超过 poll_failed_times 次没有回复设备，则设备的网络状态变为父节点丢失 NET_LOST

5.3 低功耗设备特有电池采样配置参数

- a. cap_first_delay_time: 无论剩余电量多少，默认电池上电电量都为 100%，当经过

cap_first_delay_time 时间后上报的电量才为当前真实剩余电量。

- b. cap_waitting_silence_time: 这个参数保持默认，用于 sdk。
- c. cap_max_period_time: 电池 ad 采样周期，一般和心跳周期相同
- d. cap_max_voltage: 最大采样电压
- e. cap_min_voltage: 最小采样电压

5.4 低功耗设备特有电池上报配置参数

- a. type: 电池类型为可充电电池还是干电池
- b. level: 低功耗设备默认选择空闲时采集电压
- c. report_no_limits_first: 第一次默认上报 100%电量到第二次准确上报之间的电量百分比差没有限制
- d. limits: 每两次电量上报的电量百分比差的限制（例如如果设置 20%的限制，如果电量百分变化为 30%，就会分 20%和 10%两次上报）
- e. ext_limits: 外部限制用户自选，默认为空

5.5 电池电量电压映射结构体 battery_table_t

- a. 每 10%电量变化之间默认为线性关系，一般映射表最大电压值与最大采样电压 cap_max_voltage 一致，最小电压值与最小采样电压 cap_min_voltage
- b. 低功耗设备采用电池的方式供电，电池电压的 AD 采集可以通过 VDD 也可以通过 GPIO。针对 MG21 平台，有从 GPIO 获取电压不能超过 2.42V 的限制，对于 MG13 和泰凌 8258 平台没有这个限制。因此对于 ZS3L 模组从 GPIO 采集电压，ADC 管脚必须要分压。限位值是 1.0V~2.4V，建议采用 1.2V~2.4V。

5.6 功耗注意事项

- a. 温湿度传感器不带 LCD 在网默认电流在 19.00uA.
- b. 温湿度传感器带 LCD 在各类情况下相对于不带 LCD 电流要高出 9uA
- c. 经测试温湿度传感器功耗电流在 19-20uA

5.7 sht40 温湿度转换关系

- a. 参考 sht40 官方 datasheet 可以得知，I2C 通过 6 字节将温湿度值发往模组。温度的值为 2 字节参数+1 字节 CRC 校验。湿度的值为 2 字节参数+1 字节 CRC 校验。

Humidity and temperature data will always be transmitted in the following way: The first value is the temperature signal (2 * 8-bit data + 8-bit CRC) the second is the humidity signal (2 * 8-bit data + 8-bit CRC).

图 6 sht40 官方文档数据

因此可以代码中可以看出温湿度分别从 buff 中提取出 2 字节温度值和 2 字节湿度值

buff[0]为温度值高 8 位，buff[1]位温度值低 8 位，buff[2]为温度值 CRC 校验

buff[3]为湿度值高 8 位，buff[4]位湿度值低 8 位，buff[5]为湿度值 CRC 校验

```
temp = ((uint16_t)buff[0] << 8) | buff[1];
```

```
hum = ((uint16_t)buff[3] << 8) | buff[4];
```

- b. 分别得到 2 字节温度值和 2 字节湿度值，再参考官方文档的公式可以计算出十进制的值

$$RH = \left(-6 + 125 \cdot \frac{S_{RH}}{2^{16} - 1} \right) \% RH \quad (1)$$

$$T = \left(-45 + 175 \cdot \frac{S_T}{2^{16} - 1} \right) ^\circ C \quad (2)$$

图 7 sht40 官方文档温湿度和温度计算公式

因此可以代码中可以看出转化后的温湿度，这里对得到的温度值和湿度值分别放大了 1000 倍，是为了在换算过程中减少浮点运算，将浮点转化为整型。

```
hum_hund = ((hum * 125 * 100) / 65535) * 10 - 6000;
```

```
temp_milliC = ((temp * 175 * 100) / 65535) * 10 - 45000;
```

- c. 为了在 LCD 显示实际的点分十进制温度值和百分比制的湿度值，在 LCD 显示的时候，只需要将放大的 1000 倍再除掉即可。