

配置参数结构

2025年5月13日 9:07

全局变量

```
/* 全局变量 */
typedef struct
{
    param_block_t *pConfig;    /* Current configuration */
    app_t;
    extern app_t app;
}
```

全局结构体

```
typedef struct
{
    uint32_t flag;
    uint32_t start_count;

    uint32_t HardFault_error_bit;
    uint32_t NonManage_error_bit;
    uint32_t BusFault_error_bit;
    uint32_t UsageFault_error_bit;

    param_block_t param_Config;

    uint32_t flash_error_count;
    uint32_t check_sum;
    System_Para_t;
}
```

param_Config 参数块
pConfig 参数块

```
typedef struct
{
    run_t s;    /* 基本配置 */
    dwt_config_t dwt_config;    /* 信道配置 */
    tag_addr_slot_t knownTagList[MAX_KNOWN_TAG_LIST_SIZE];    /* 171字节 19 * 171 / 9 */
    __attribute__((packed)) param_block_t;
}
```

_t 表示它是一个类型定义，通常意味着它是通过 typedef 定义的别名

baseConfigtwr 基础配置

```
__packed typedef struct
{
    uint16_t antRx;    /* 天线接收(接受) */
    uint16_t antTx;    /* 天线发送(发送) */
}baseConfig_t;
```

sfConfig twr sf配置

```
__packed typedef struct
{
    uint16_t slotPeriod;    /* 单个槽周期 */
    uint16_t numSlots;    /* 槽数量 */
    uint16_t sPeriod_ms;    /* 超级帧周期 = 单个槽周期 * 槽数量 */
    uint16_t tag_replyoly_us;    /* 标签发送Poll->接受Resp延迟时间 */
    uint16_t tag_pollToFinalTx_us;    /* 标签发送Poll->发送final延迟时间 */
}sfConfig_t;
```

userConfig 用户配置

```
__packed typedef struct
{
    uint16_t address;    /* 节点地址 */
    uint8_t role;    /* 节点角色 */
    uint8_t workmode;    /* 工作模式 */
    uint16_t ror_pdoa_mode;    /* 节点角色 */
    uint16_t ror_pdoa;    /* 节点角色 */
    uint16_t nodeFilter;    /* 节点滤波 */
    uint16_t tagnumSlots;    /* 默认标签槽数量10个,最大32个 */
    float kalman_0;    /* 滤波参数0 */
    float kalman_R;    /* 滤波参数R */
    float para_A;    /* 标定参数A */
    float para_B;    /* 标定参数B */
    uint16_t Pn_mod;    /* 功率设置 */
}userConfig_t;
```

s_pdoa pdoa基本配置

```
typedef struct {
    uint16_t addr;    /* 基站地址 Node's address */
    uint16_t panId;    /* 系统网络 System PanID */
    uint16_t pdoaOffset_deg;    /* Calibration: the Phase Differences Offset */
    int16_t rngOffset_ms;    /* Calibration: the Ranging offset */
    uint16_t pdoaTempCoeff_pdoa;    /* reserved for future */
    uint8_t rbcEn;    /* Enable Range bias correction */
    uint8_t phaseCorEn;    /* Use correction curve to use in calculation PDOA->X-Y */
    uint8_t smartTxEn;    /* Enable SmartTx power */
    uint8_t reportLevel;    /* 0 ~ not output, 1-350N, 2-limited */
    uint16_t acc_stationary_ms;    /* Number of milliseconds to decide the device is not moving */
    uint16_t acc_moving_ms;    /* Number of milliseconds to decide the device is moving */
    uint16_t acc_threshold;    /* The threshold for X-Y axis, in milli-6, to separate instantaneous movement to stationary */
    uint16_t rDelay_us;    /* Node's delay between end reception of UWB blink and start transmission of UWB Ranging Config message */
    uint16_t motionFilter;    /* 是否开启滤波 */
    uint16_t user_ced;    /* 格式选择 */

    uint8_t Dlist;    /* 发现列表数量 */
    uint8_t Klist;    /* 绑定列表数量 */
    uint8_t UserRefreshRate;    /* 端口刷新速率 */
    uint8_t reserve;    /* 保留 */

    uint16_t antRx_A;    /* antenna delay values for current MASTER */
    uint16_t antTx_A;    /* antenna delay values for current MASTER */
    uint16_t antRx_B;    /* antenna delay values for current SLAVE */
    uint16_t antTx_B;    /* antenna delay values for current SLAVE */
    sfConfig_t pdoa sfConfig_pdoa;    /* System configuration: SuperFrame description */
    __attribute__((packed)) run_pdoa_t;
}
```

```
__packed typedef struct
{
    uint16_t slotPeriod;    /* Slot period (time for one tag to range) */
    uint16_t numSlots;    /* Number of slots used in the SuperFrame */
    uint16_t sPeriod_ms;    /* SuperFrame period in ms */
    uint16_t tag_replyoly_us;    /* wait4response delay after end of Poll for the tag, us */
    uint16_t tag_pollToFinalTx_us;    /* PollToFinalTx delay, us */

    /* below can be removed if the d2k command, i.e. "add all discovered tags to known list automatically", is not used */
    uint16_t tag_nput;    /* Used to pass a "moving" refresh rate to all tags in the "d2k" command */
    uint16_t tag_nslow;    /* Used to pass a "stationary" refresh rate to all tags in the "d2k" command */
    uint16_t tag_Mode;    /* Used to pass a common "mode" parameter to all tags in the "d2k" command */
}sfConfig_t_pdoa;
```

s基本配置

```
typedef struct
{
    baseConfig_t baseConfig;    /* twr 基础配置 */
    sfConfig_t sfConfig;    /* twr sf配置 */
    userConfig_t userConfig;    /* 用户配置 */
    run_pdoa_t s_pdoa;    /* pdoa基本配置 */
    __attribute__((packed)) run_t;
}
```

dwt_config信道配置

```
/* This Enum holds the index for factor calculation. */
typedef enum
{
    /* Structure typedef: dwt_config_t */
    /* Structure for setting device configuration via dwt_configure() function */
    /* */
    __packed typedef struct
    {
        uint8_t chan;    /* Channel number (5 or 9) */
        uint8_t txPreambleLength;    /* DWT_PLEN_64, DWT_PLEN_4096 */
        uint8_t rxPAC;    /* Acquisition Chunk Size (Relates to RX preamble length) */
        uint8_t txCode;    /* TX preamble code (the code configures the PRF, e.g. 9 -> PRF of 64 MHz) */
        uint8_t rxCode;    /* RX preamble code (the code configures the PRF, e.g. 9 -> PRF of 64 MHz) */
        uint8_t sfType;    /* SFD type (0 for short IEEE 8-bit standard, 1 for DW 8-bit, 2 for DW 16-bit, 3 for 4z 8BPRF) */
        uint8_t dataRate;    /* Data rate (DWT_BR_850K or DWT_BR_60M) */
        uint8_t phrMode;    /* PHR mode (0= standard DWT_PHRMODE_STD, 0x3 - extended frames DWT_PHRMODE_EXT) */
        uint8_t phrRate;    /* PHR rate (0= standard DWT_PHRRATE_STD, 0x1 - at dataRate DWT_PHRRATE_DIA) */
        uint16_t sfdTo;    /* SFD timeout value (in symbols) */
        uint8_t stsMode;    /* STS mode (no STS, STS before PHR or STS after data) */
        dwt_sts_lengths_e stsLength;    /* STS length (the allowed values are listed in dwt_sts_lengths_e) */
        uint8_t pdoaMode;    /* PDOA mode */
    } dwt_config_t;
```

```
/* This Enum holds the index for factor calculation. */
typedef enum
{
    DWT_STIS_LEN_32 =0,
    DWT_STIS_LEN_64 =1,
    DWT_STIS_LEN_128 =2,
    DWT_STIS_LEN_256 =3,
    DWT_STIS_LEN_512 =4,
    DWT_STIS_LEN_1024 =5,
    DWT_STIS_LEN_2048 =6
} dwt_sts_lengths_e;
```

knownTagList tag地址集合

```
typedef struct
{
    uint16_t slot;
    __packed union {
        uint8_t addrShort[2];
        uint16_t addr16;
    };
    __packed union {
        uint8_t addrLong[8];
        uint64_t addr64;
    };
    uint16_t multiFast;
    uint16_t multiSlow;
    uint16_t mode;    /* INU = bit 0 */
    __packed union {
        uint8_t req;
    };
    __attribute__((packed))
    tag_addr_slot_t;
}
```