守护进程通信协议

**REVISION HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date | Section | Page | Line | Description |
| 1.0.0 | 4/8/2019 |  |  |  | 新增协议概述 |

# 心跳协议概述

守护进程作为TCP服务端，ip地址为：127.0.0.1，端口号为：12539；

作为UDP服务端，ip地址为：127.0.0.1，端口号为：12540。

TCP，UDP客户端周期向守护进程发送心跳消息，且消息为客户端向服务端单向广播模式。客户端发送周期、服务端监测周期均设定为TC=10s，超时时间(timeout)为2TC = 20s。心跳判断规则：如果最近的心跳消息的接收时间早于now – timeout，可判断失败。此后守护进程负责杀死进程，并重启应用。

## 协议格式

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Header[21] | | | Fields[N] | | | | Parity | |
| START\_BYTE[1] | LEN[2] | FLAG[18] | FIELD[0] | FIELD[1] | …… | FIELD[N-1] | PKT[1] | DATA[1] |
| **0xcd** |  |  |  |  |  |  | **0x2f** |  |

### 协议头说明

//数据传输顺序为低字节在前，高字节在后。

//目前版本不启用校验单元

Format: Header + Fields （+ Parity）;

Header: START\_BYTE + LEN + FLAG;

START\_BYTE: 固定为0xCD;

FLAG: ip[4] + port[2] + start\_time[8] + pid[4];

Ip：心跳消息的源ip地址信息；

Port：心跳消息的源port端口号；

Start\_time：心跳消息所在进程的启动时间（从设备启动到进程启动的微妙数,参考[获取方法](#_获取当前时间戳说明)）；

Pid：心跳消息所在进程的Pid号（参考[获取PID方法](#_获取当前线程PID说明)）。

PARITY:

PKT:固定为0x2f;

DATA:除了START\_BYE和DATA字段以外，数据包中所有字节的异或结果。

LEN: 数据包的总长度，即包含FIELD[0]到FIELD[N-1]，并加上Parity数据（如果启用了校验）的总和（注意并不包含Header[21]）。

### 协议数据段说明

Fileds字段具体定义如下：

|  |  |  |
| --- | --- | --- |
| Fileds[2] | | |
| App\_type[2] | Payload[0] |  |
| 0x0001:fsu\_client  0x0002: Wire88FSUManager  0x0003:  0x0004: | 目前版本负载内容为空 |  |

### 获取当前时间戳说明

#include <sys/time.h>

static const int kMicroSecondsPerSecond = 1000 \* 1000;

Int64\_t getNowTimestamp()

{

struct timeval tv;

gettimeofday(&tv, NULL);

int64\_t seconds = tv.tv\_sec;

return (seconds \* kMicroSecondsPerSecond + tv.tv\_usec);

}

### 获取当前线程PID说明

注意：在main()调用此接口以获得主线程PID号

#include <sys/syscall.h>

int getCurrentThreadID()

{

int pid = static\_cast<pid\_t>(syscall(SYS\_gettid)); //获取当前线程的pid（系统调用，频繁使用，效率低下）

return pid ;

}