## 1、网络通信机制

网络通信协议暂定为TCP/IP协议，数据传输使用字节数组。SC280采用小端格式，而本系统采用大端格式，所以在通信过程中需要注意数据存储格式。

### 1.1 打开网络设备

在用户需要连接网络设备的时候，首先需要调用open()函数和网络设备建立连接。那么在open()接口函数中我们需要启动子线程，并在线程中创建socket对象，建立TCP/IP连接。请求连接的过程如下所示：



**图1 打开NET过程示意图**

### 1.2 数据包的产生

数据包里包含以下几个部分：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| magic | version | type | block | length | offset | minid | data[1] |

其中前7个部分是数据帧头，而后面的是用户数据。这里主要需要控制改变的是type、block、minid以及用户数据这几个部分。所以在生成数据包的过程中，仅对这几项做控制。

首先我们根据协议针对不同的命令编写相应的数据包；然后通过工厂方法模式根据用户需求生成相应数据包。

### 1.3 数据发送

应用程序可以通过调用数据包的send()函数发送数据。这里将send()函数封装在了数据包对象当中。调用send()之后，程序从Socket中获取输出流。然后向网络输出流中写入我们要发送的数据。

发送数据过程如下：

1. 通过用户操作判断需要执行的命令；
2. 根据协议找到需要生成的数据包；
3. 将数据包的帧头写入输出缓冲区；
4. 如果此数据包含有数据区，则在帧头后面紧跟用户数据；
5. 最后将输出缓冲区的数据统一转换成字节数组，并通过网络发送。

这样数据包的发送就完成了。

### 1.4 网络数据接收

和数据发送一样，这里将接收函数recvDataPack封装到数据包对象当中。当我们需要去接收一个数据包的时候，直接调用数据包的recvDataPack()函数。数据的接收会以阻塞的方式进行，设置网络超时时间为7秒。

数据的接收及解析过程如下：

1. 调用recvDataPack()函数后，获取Socket输入流；
2. 设置一个2M的接收缓存，用于接收网络上的数据；
3. 从接收缓存中查找4字节的帧头标识（0x695A695A），如果未找到则舍弃；
4. 如果找到帧头，则记录下位置，并判断数据帧头是否接收完整。如完整，则对帧头进行解析，反之则舍弃此包。
5. 在数据帧头解析完毕后，可获得数据长度字段，按照此字段将整包数据打包存入接收数据包对象中完成接收。
6. 将数据包的用户数据部分传给上层，供应用层使用。

**图2 网络接收流程图**

### 1.5 关闭网络设备

当长时间不需要数据传输时，和关闭系统之前，建议调用close()函数关闭网络连接。另外，在网络连接过程中，如果出现任何故障，系统也会自动关闭网络连接。

## 2、网络通信时间消耗

### 2.1 通过网络获取一张图片

（1）发送数据包：

|  |  |  |
| --- | --- | --- |
| 字段 | 值 | 备注 |
| type | 1 | 需要返回 |
| block | 5000 | 超时5000ms |
| data | 无 | 请求帧，不需要数据 |

无数据段，长度为28字节。

（2）接收数据包：

|  |  |  |
| --- | --- | --- |
| 字段 | 值 | 备注 |
| data | len (4B) | 图像大小，单位：字节 |
| width (4B) | 图像宽度 |
| height (4B) | 图像高度 |
| bpb (4B) | 每个像素占字节数 |
| reserve (84B) | 保留 |
| image | 图像数据 |

接收一张640×480的图片，数据段大小为307300。整个数据包大小为307328字节。

从发送命令到接受完整张图片一共需要170—210ms。

### 2.2 通过网络获取温度

（1）发送数据包：

|  |  |  |
| --- | --- | --- |
| 字段 | 值 | 备注 |
| type | 1 | 需要返回 |
| block | 50000 |  |
| data | 1 (4B) | ADT75\_GET\_TEMPVALUE，获取温度子命令 |

数据段长度为4个字节，表示温度子命令。整个包长度为32字节。

（2）接收数据包：

|  |  |  |
| --- | --- | --- |
| 字段 | 值 | 备注 |
| data1 | 1 (4B) | ADT75\_GET\_TEMPVALUE子命令 |
| interger (4B) | 温度的整数部分，若35.6度，则为35 |
| float (4B) | 温度的小数部分，若为35.6度，则为6 |
| data2 | 0 (4B) | 读取温度失败 |

数据段为12个字节，表示实际的温度值。整个包长度为40字节。

从发送命令到读取完一共需要57—86ms。

## 3、应用程序内存消耗

由于Android运行在Linux系统之上，应用程序在运行过程中需要在底层调用Linux共享库，而许多共享库有很多程序都会使用，但其实在Linux系统的整个生命周期中只加载过一次，而对内存的消耗是微乎其微的。

所以，在Android应用程序中，我们通常用USS（Unique Set Size），即进程独自占用的物理内存来作为内存消耗的参考值。

本应用程序所占内存大约为16644KB。