NEC808 PC interface

Peter Blockley

October 3, 2003

1 Interface

The NEC808, hereafter referred to as "the phone", attaches to the computer via a USB cable. A WDM(Windows Driver Model) is installed, called "NEC 808 USB Controller". The "NEC e808 PC file transfer" software communicates with the WDM via a virtual COM port, which in turn communicates with the phone. The virtual COM port may be accessed similar to the physical com port(namely through API calls).

The phone transfers information with packets of maximum size 20480 bytes. Each packet contains a header, which details the packet size and indicates if it is the last packet in the sequence.

To start a transmission the PC sends the following to the virtual COM port:

```
HEX: 80 00 07 10 00 FF FF
```

The phone replies with:

```
HEX: A0 00 07 10 00 50 00
```

The PC may now transfer information to the phone. This may be a file, a contact or a calender item. Files(wma, mid, ect) are transferred using VNOTE(description to follow). The files are first encoded in BASE64, a method of transmitting binary data as printable text, before being transmitted to the phone and converted back into binary.

when all the packets have been transmitted, the phone responds with:

```
HEX: A0 00 03
```

The PC responds with:

```
HEX: 81 00 03
```

The phone responds with:

```
HEX: A0 00 03
```

The session is now finished.

2 First Packet Headers

2.1 Ringtone

The header of the first packet when sending a ringtone is:
`HEX: 02 50 00 01 00 21 00 6E 00 65 00 63 00 5F 00 6D`
`HEX: 00 65 00 6C 00 6F 00 64 00 79 00 2E 00 76 00 6E`
`HEX: 00 74 00 00 48 4F DC`

The first three bytes tell the phone:
`HEX: 02` - sending multiple packets (0x82 for single packet)
`HEX: 50 00` - this packet is 20480 bytes long(0x50 0x00 is 20480 in decimal)

the middle of the header tells the phone your uploading a melody(ringtone) in VNOTE format("..!.n.e.c._.m.e.l.o.d.y...v.n.t..").

The last three bytes tell the phone:
`HEX: 48` - sending multiple packets(0x49 for single packet)
`HEX: 4F DC` - packet length less 36 bytes (20480 bytes - 36 bytes)

2.2 Audio Files

The header of the first packet when sending an audio file is:
`HEX: 02 50 00 01 00 1F 00 6E 00 65 00 63 00 5F 00 61`
`HEX: 00 75 00 64 00 69 00 6F 00 2E 00 76 00 6E 00 74`
`HEX: 00 00 48 4F DE`

The first three bytes tell the phone:
`HEX: 02` - sending multiple packets (0x82 for single packet)
`HEX: 50 00` - this packet is 20480 bytes long(0x50 0x00 is 20480 in decimal)

the middle of the header tells the phone your uploading an audio in VNOTE format("....n.e.c._.a.u.d.i.o...v.n.t..").

The last three bytes tell the phone:
`HEX: 48` - sending multiple packets(0x49 for single packet)
`HEX: 4F DE` - packet length less 34 bytes (20480 bytes - 34 bytes)

2.3 Other file types

The header of the first packet when sending a "other file type" is:
```
HEX: 82 22 9D 01 00 1D 00 6E 00 65 00 63 00 5F 00 6D
HEX: 00 69 00 73 00 63 00 2E 00 76 00 6E 00 74 00 00
HEX: 49 22 7D
```

The first three bytes tell the phone:
`HEX: 82` - sending single packet(`HEX: 02` for multi-packet)
`HEX: 22 9D` - this packet is 8861 bytes long

the middle of the header tells the phone your uploading a misc(other file type) in VNOTE format("....n.e.c._.m.i.s.c...v.n.t..").

The last three bytes tell the phone:
`HEX: 49` - sending single packet (0x48 for multi-packet)
`HEX: 22 7D` - packet length less 32 bytes (20480 bytes - 32 bytes)

2.4 Calender

Calender items are transmitted in a single packet using VCALENDAR format.
The header for a calender item is:
```
HEX: 82 01 1F 01 00 13 00 63 00 61 00 6C 00 2E 00 76
HEX: 00 63 00 73 00 00 49 01 09
```

The first three bytes tell the phone:
`HEX: 82` - There is only 1 packet
`HEX: 01 1F` - The packet is 287 bytes long

The middle of the header tells the phone a VCS file is being transmitted("....c.a.l...v.c.s..").

The last three bytes tell the phone:
`HEX: 49` - sending single packet
`HEX: 01 09` - packet length less 22 bytes (20480 bytes - 22 bytes)

2.5 Contacts

Contacts are transmitted in a single packet using a VCARD format. The header for a contact packet is:

```
HEX: 82 00 E0 01 00 11 00 70 00 62 00 2E 00 76 00 63
HEX: 00 66 00 00 49 00 CC
```

The first three bytes tell the phone:
`HEX: 82` - There is only 1 packet
`HEX: 00 E0` - The packet is 224 bytes long

The middle of the header tells the phone a VCF file is being transmitted(“....p.b...v.c.f..”).

The last three bytes tell the phone:
`HEX: 49` - sending single packet
`HEX: 00 CC` - packet length less 20 bytes (20480 bytes - 20 bytes)

2.6 Video

The header of the first packet when sending a “video file” is:

```
HEX: 02 50 00 01 00 1F 00 6E 00 65 00 63 00 5F 00 76
HEX: 00 69 00 64 00 65 00 6F 00 2E 00 76 00 6E 00 74
HEX: 00 00 48 4F DE
```

The first three bytes tell the phone:
`HEX: 02` - sending single packet(`HEX: 02` for multi-packet)
`HEX: 50 00` - this packet is 20480 bytes long

the middle of the header tells the phone your uploading a video in VNOTE format(“....n.e.c._.v.i.d.e.o...v.n.t..”).

The last three bytes tell the phone:
`HEX: 48` - sending multiple packets (0x49 for single-packet)
`HEX: 4F DE` - packet length less 34 bytes (20480 bytes - 34 bytes)

2.7 Images

The header of the first packet when sending an image is:

```
HEX: 02 50 00 01 00 1B 00 6E 00 65 00 63 00 5F 00 69
HEX: 00 6D 00 67 00 2E 00 76 00 6E 00 74 00 00 48 4F
HEX: E2
```

The first three bytes tell the phone:
`HEX: 02` - sending single packet(`HEX: 02` for multi-packet)
`HEX: 50 00` - this packet is 20480 bytes long

the middle of the header tells the phone your uploading a video in VNOTE format(“....n.e.c._.i.m.g...v.n.t..”).

The last three bytes tell the phone:

HEX: 48 - sending multiple packets (0x49 for single-packet)
HEX: 4F DE - packet length less 30 bytes (20480 bytes - 30 bytes)

2.8 Voice

The header of the first packet when sending an image is:
HEX: 02 50 00 01 00 1F 00 6E 00 65 00 63 00 5F 00 76
HEX: 00 6F 00 69 00 63 00 65 00 2E 00 76 00 6E 00 74
HEX: 00 00 48 4F DE

The first three bytes tell the phone:
HEX: 02 - sending single packet(HEX: 02 for multi-packet)
HEX: 50 00 - this packet is 20480 bytes long

the middle of the header tells the phone your uploading a video in VNOTE format("....n.e.c._.v.o.i.c.e...v.n.t..").

The last three bytes tell the phone:
HEX: 48 - sending multiple packets (0x49 for single-packet)
HEX: 4F DE - packet length less 30 bytes (20480 bytes - 30 bytes)

3 Data Transfer Modes

3.1 VNOTE

After the header, the first packet contains the VNOTE information regarding file name, size, ect. Each line is terminated with HEX: 0D 0A(CR LF)

```
BEGIN:VNOTE
VERSION:1.1
SUMMARY;CHARSET=UTF-8;ENCODING=QUOTED-PRINTABLE:<name>
X-NEC-FILENAME;CHARSET=UTF-8;ENCODING=QUOTED-PRINTABLE:<file_name>
X-NEC-FILESIZE:<file_size>
X-NEC-INDEX:
BODY;ENCODING=BASE64:
```

where `name` is the name you would like to associate with the file(normally the `file_name` without the extension), `file_name` is the file name and `file_size` is the size in bytes of the binary file(before BASE64 encoding).

The BASE64 encoded data is then transmitted to the phone, using the remaining space in the first packet.

When a packet of a multi-packet transmission has been transmitted, the phone responds with:

```
HEX: 90 00 03
```

If multiple packets of 20480 bytes are required to transmit the information, then the next packet will have header:

```
HEX: 02 50 00 49 4F FD
```

The last packet will have header:

```
HEX: 82 1B BE 49 1B BB
```

This tells the phone:
```
HEX: 82 - This is the last packet
HEX: 1B BE - this packet is 7102 bytes long
HEX: 49 - This is the last packet
HEX: 1B BB - packet length minus 3 (7102 bytes-3 bytes)
```

The last packet will have footer:

```
HEX: 0D 0A 45 4E 44 3A 56 4E 4F 54 45 0D 0A
```

which is simply "END:VNOTE"

3.2 VCARD

The VCARD is transmitted in plain text. Each line is terminated with HEX: 0D 0A(CR LF) the same as a windows text file. These are some of the allowable fields. Not all fields are required

```
BEGIN:VCARD
VERSION:2.1
N;CHARSET=UTF-8;ENCODING=QUOTED-PRINTABLE:<name>;;;;
FN;CHARSET=UTF-8;ENCODING=QUOTED-PRINTABLE:<name>
TEL;HOME;VOICE:<tel1>
TEL;CELL:<tel2>
TEL;WORK;VOICE:<tel3>
EMAIL;INTERNET:<email1>
EMAIL;HOME:<email2>
EMAIL;WORK:<email3>
END:VCARD
```

3.3 VCALENDER

The VCALENDER is transmitted in plain text. Each line is terminated with HEX: 0D 0A(CR LF) the same as a windows text file. Here is an example:

```
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
DTSTART:20031006T230000Z
DTEND:20031006T230000Z
DESCRIPTION;CHARSET=UTF-8;ENCODING=QUOTED-PRINTABLE:hair cut
SUMMARY;CHARSET=UTF-8;ENCODING=QUOTED-PRINTABLE:hair cut
CATEGORIES:
RRULE:
END:VEVENT
END:VCALENDAR
```