Controller Network Protocol (Rev. 1.1)

Length	Source	Destination	Message	Options....	Check-A	Check-B

Where :

- Length is the number of bytes in the message, not counting itself or the Checksum Bytes.
- Source is the port number of the sender of the message.
- Destination is the port number of the recipient of the message, ID 255 is reserved for broadcast messages
- Message is the code to indicate the contents of the message.
- Options are unique to each message and can be between zero and 30 bytes.
- Check-A is the 1's complement addition of all bytes up until the checksum. Check-B is the 1's complement addition of all the Check-A values as they are calculated (8-bit Fletcher Checksum).

SLIP Framing is used over the network even though IP format is not used. The framing character is $C0. Occurrences of $C0 are replaced with $DBDC while occurrences of $DB are replaced with $DBDD. Each packet starts and ends with the framing character ($C0).

Port Number Assignments

0 = Controller
1-250 = Remote Units
251-254 = Reserved
255 = Broadcast

Universal Messages

0 = Short Network Time Message, 1 byte, deka-minutes since midnight. This is a broadcast message normally sent at 10, 20, 30, 40, and 50 minutes past the hour.
1 = Controller requests unit ID, all units return with message 2.
2 = Unit ID, first byte is unit type, second byte is software version. Some units may return an additional byte that indicates the starting address of their programming block.
3 = Set Unit's internal clock, 3 bytes. 1st byte is deka-minutes since midnight, 2nd byte is minutes mod 10, 3rd byte is seconds, this also sets the unit's "programmed" status bit, if it has one.
4 = Request status, if there is no option byte then unit returns with default number of status bytes, else returns with specified number of status bytes. All units return with a number 5 message, format varies.

Recommended Practices

SLIP framing allows accurate detection of packet boundaries. If the length field does not equal the number of characters (after de-framing) in the packet minus 3, then the packet is invalid and must be ignored.

Other than the Universal messages, message numbers should be diverse between types of nodes. Each node needs to reject any message that appears to be addressed to it, but not valid for that type of node. This reduces the chances of a misdirected message.

Noise Rejection

To test noise rejection of this packet format I used a stream of 3.78432 Billion random characters. This represents 1 years worth of characters at 1200 baud. 1.6 Million valid frames were detected (a string of characters <= 36 characters long between two framing characters). Of those frames, 5765 of those passed the first test, where the length byte was valid for the number of characters in the frame. Of those, no frames passed the checksums.

Improving Error Detection

Verification information can be inserted into extra option bytes to improve detection of a corrupted packet beyond what the checksums can detect. For instance, in a typical 6 byte long packet used to dispense material where the quantity is specified as the option, with a valid range of 1 to 31, doing error detection testing resulting in :

```
Single Bit changes not detected or harmless = 0.0000%
Double Bit changes not detected or harmless = 0.0000%
Triple Bit changes not detected or harmless = 0.0114%
Quad   Bit changes not detected or harmless = 0.0030%
```

Adding another byte which is the quantity shifted left once resulted in these values :

```
Single Bit changes not detected or harmless = 0.0000%
Double Bit changes not detected or harmless = 0.0000%
Triple Bit changes not detected or harmless = 0.0023%
Quad   Bit changes not detected or harmless = 0.0010%
```

For very critical messages you could even use a 32 bit CRC in the option bytes to virtually eliminate the possibility of an undetected error. These tests do not take into account packets that would be rejected by missing bytes or ASYNC framing errors caused by noise.

Encapsulated CNP (ECNP)

CNP can be sent over the Internet encapsulated in UDP packets. TCP makes no sense since CNP is a datagram oriented protocol like UDP, not a byte stream like TCP. SLIP framing is not used in an ECNP packet, the UDP payload is the original binary, non-framed, non-escaped packet. The combination of the CNP checksum and the UDP checksum reduces the error rate further.

A suggested use of ECNP is to have a registered port on a system. Incoming packets (via UDP) would be sent out through a serial port on the system. Responses to those packets would be returned to the IP address and UDP port number of the sender. UDP Port 2858 has been assigned by IANA (Internet Assigned Numbers Authority) for the registered port.

CNP Over Ethernet

Although not tested, it should be possible to use ethernet to send/receive CNP packets. For instance, a CS8900 or RTL8019 could be connected to a micro-controller to provide a fast and low cost interface. Using CNP instead of UDP/IP would save on code and storage requirements. Like ECNP, no SLIP framing would be used.

Ethernet adds the complication of the hardware address. This can be handled either by always using the ethernet broadcast MAC address, or by using the broadcast address only to initially get the MAC address of each device on the network. For instance, the ID request could be sent as a broadcast, the device having the port address in the message would respond to the controller. From the response the controller would pick up the device's MAC address and save it for all future communications.

In order to distinguish a CNP message from IP or other protocol messages, a unique "Ethertype" is required. For instance $0800 is used for IP and $0806 is used for ARP. CSC recommends using an ethertype of $09A4 (Decimal 2468).