Rick DeV.
Mike S.
Mike E.
Ed. W
Paul K.
Andy P.
Jill P.
Richard H-G.

ACKØ & ACK1 should not be valid IR addresses
HDL address → 11_{10}

DATALINK COMMUNICATIONS PROTOCOL - Version 0.90

Mike Saari 9/8/83
File: protocol.tp2

Murphy strikes back! This is the latest revision of the datalink protocol. Version 1.0 is no longer the most current revision.

Summary of changes:
Power-up sequence specified. IR carrier timeout reduced to 1.28 seconds. CONFIGURE DATALINK command added. "NOT CONFIGURED" flag added to QUERY responses. The datalink is now completely decoupled from channel, process, and command #'s. True/false now represented by 1 or 0. GET LAST RESPONSE returns 5 chars, not 4. RESET LINK no longer affects the message buffer or the channel number. Messages require both proc# and cmd#, otherwise invalid message is set. GET LINK VERSION# added.

This document will deal strictly with host datalink (H.D.) protocols. Commands passing between the Apple and Topo are covered in a separate document.

Packet Format

All messages are sent over the IR link in packets. The packet format itself has not been changed, i.e. one bit every 512 usec, starting with 5 on's and 1 off, then characters coded as 8 data bits (LSB first) followed by 1 parity bit - odd parity. All packets in the system are always 8 characters, expect for short ACK's which are only a single character. The 8 characters in the packet are ordered as follows:

ch#, proc#, cmd#, d,d,d,d, cksum .

D,d,d,d are 4 characters of data; ch# is the channel number to indicate which Topo(s) should respond; proc# is the on-board process number; cmd# is the command for the given process; and cksum is the twos complement of the sum of all the previous characters.

All packetizing, retries, and IR error handling is handled by the H.D.

8-bit Data and Nibbling

All data in the system, whether on the Apple or onboard Topo, consists of full 8-bit words. There is a problem, however, in that most serial cards deal with 7-bit ASCII, and also trap certain control codes. To avoid this, all data sent between the Apple and H.D. will be broken up into 4-bit nibbles, sent through the serial card, and then reassembled. (The full 8-bits are sent over the IR.) The 4-bit nibbles are encoded as ASCII 0-9 and A-F. For example, data 10100011 would be sent as A3 (hex 41, hex 33). A letter G (hex 47) would be sent as 47 (hex 34, hex 37).

Apple/H.D. Command Syntax

Handshaking commands between the Apple and H.D. are not broken up into nibbles. All commands and responses are standard ASCII characters, excluding 0-F and excluding all control codes. The actual range of values is hex 50-7F.

Apple/H.D. Serial Bus Communication

Although the baud rates are setable on both the serial card and the H.D., the standard configuration is: 9600 baud, no parity, 8 data bits, and 1 stop bit.

Apple/H.D. Handshaking

A partial handshaking system is used between the Apple and the H. D. Before sending any message or command to the H.D., the Apple must first send a

QUERY (ASCII Q).

The H.D. must respond with one of several answers to indicate readiness to receive the message. The required response time is within 5 ms + 1 character time (dependent on the baud rate). (This protocol will run at 9600 baud, which gives a character time of 1 ms.) The possible responses are:

```
   Bit 4 - ERROR - not configured (reset upon receipt of
                   CONFIGURE DATALINK and CONFIGURE PACKET)
       3 - BUSY  - Apple may not send more data yet
       2 - MESSAGE WAITING (from Topo to Apple)
       1 - ERROR - Topo not responding
       0 - ERROR - invalid message from Apple to H.D.
 all off - READY - OK to send
```

(The H.D. will accept data even if MESSAGE WAITING is set. However, the Apple will generally read the message first.)

The response is a single character, where the indicated bit is active. The three high-order bits are always 111, so responses are E0-FF (ASCII lowercase characters). Thus MESSAGE WAITING is 11100100, or hex E4, and BUSY is 11101000 or hex E8 .

The H.D. may interrupt a long message coming in by sending a BUSY back to the Apple, which must then stop sending and QUERY until a READY is received. (The Apple always checks before sending any character to see if a BUSY has been received.)

Apple/H.D. Command Formats

The standard message format from Apple to H.D. is:
MESSAGE START (ASCII S)
Topo process number (nibbled)
Topo command number (nibbled)
d,d,d ... d (nibbled)
MESSAGE END (ASCII Z).

This is used for every message which is to be sent from the Apple to Topo. If the H.D. detects an extra MESSAGE START, an extra MESSAGE END, a MESSAGE END before proc# and cmd# have been received, a framing error or other serial error, or a MESSAGE END after only half of a character (one nibble) has been sent, then the invalid message error code is set for the next query. Remaining characters at the end of a message, which fall short of filling a packet, are sent out in a final packet padded with trailing spaces (hex 20). If the data for a message spans more than one packet, then the Topo process number and command number will be duplicated for each packet.

CONFIGURE PACKET (ASCII P) tells the H.D. the channel # for further packets, and a public/private flag to indicate whether to expect a return ACK. One of two commands needed to reset the "not configured" flag. The format is:
CONFIGURE PACKET (ASCII P)
channel # (nibbled)
public flag - 1=public, 0=private (nibbled)

When the public flag is set, no ACK's are expected.

CONFIGURE DATALINK (ASCII Y) tells the datalink its own IR channel #, and the format for the carrier dummy message. The IR carrier function is then enabled. One of two commands needed to reset the "not configured" flag. The format is:
CONFIGURE DATALINK (ASCII Y)
H.D. channel # (nibbled)
carrier channel # (nibbled)
carrier proc# (nibbled)
carrier cmd# (nibbled)

GET LAST RESPONSE (ASCII R) will cause the H.D. to return (0,d,d,d,d) (in nibbles) that was last received back from Topo. The "0" is a don't care character returned with the packet. It may be used in the future as a response identifier.

RESET LINK (ASCII X) cancels the current packet being attempted and resets the four lower QUERY status flags (all except "not configured").

GET LINK VERSION# (ASCII V) causes the H.D. to return two bytes (in nibbles) of the H.D.'s version number. Each byte has values 0-99. The first byte is integral version, the second is fractional version, i.e. 1.00, 2.05, 99.99 . The initial release will be 1.00, prototypes start at 0.01 .

In the middle of a message the only legal H.D. commands are QUERY and RESET LINK, otherwise just data and END-OF-MESSAGE (after READY has been received.) In fact, whenever any BUSY has been received, the only legal commands are QUERY and RESET LINK.

Power-on Sequence

The Apple should be powered-on, or reset, after the H.D. has been powered-on. This is because the Apple reset causes Toposoft to perform a CONFIGURE DATALINK and CONFIGURE PACKET. The H.D. will not send any IR until this has occurred.

ACK0,ACK1, and Saywhat Protocol

Every message received by any given Topo (except for public channels), is acknowledged by an ACK0 or an ACK1. An ACK0 "packet" is a single letter Q; an ACK1 packet is a single letter R. These must alternate to insure valid transmissions. The MSB of the channel# being transmitted indicates whether Topo should send an ACK0 or ACK1 - 0 means ACK0, 1 means ACK1. The H.D. is required to keep track of the current ACK0/ACK1 status for each private channel.

A valid ACK should be received by the H.D. within 50 ms of the end of the message being sent. If no ACK is received within that time, or a garbled transmission or the wrong ACK is detected, then a "saywhat?" command (see Message Formats document) is given, telling the Topo to repeat its last ACK. Receiving the correct ACK means to continue normally, i.e. the original message has been correctly handled. Receiving the wrong ACK means the original message was mishandled and should be resent. Receiving nothing or a garbled reply should prompt another "saywhat?". After 3 or more saywhat's without a valid reply, the appropriate QUERY error flag (Topo not responding) is set while the saywhat's continue. The error flag will only be reset after the message is eventually received, or by doing a RESET LINK.

If Topo is ever unable to determine the answer to a status request within the ACK timeout time, it simply sends the appropriate ACK with no message. The Apple may then deal with the error of "message not received". This timeout handler could change with future product revisions.

When Topo sends a message to the H.D. (generally a status requested by the Apple), the format is:

H.D. channel #,
Q/R (appropriate ACK),
0 (don't care)
d,d,d,d,
cksum.

IR Carrier

A carrier signal tells all Topos if they go out of range of the H.D. The H.D. is required to send out some command at least once every 300 msecs. If no command has been sent after 300 msecs has elapsed, the H.D. sends out a dummy message whose content is determined by CONFIGURE DATALINK. Any Topo which hears no valid commands for 1.28 seconds will PARK (abort the current motion command and flush the motion command queue).

Real-time Joystick

Since the packets are relatively short (8 characters at 2000 baud IR gives 40 ms per packet), this should work for real-time joystick response. Since short ACK's take only another 5 ms (plus computation time), it may well work with ACK's, too. The typical configuration will be to use a public channel so as to ignore any message errors, since we are better off simply sending the newest command, instead.