Sycon Multi Drop Protocol
DOC VERSION 1.4
2003-06-11
Dave Reynolds
FOR PROTOCOL VERSIONS 2 and 3

INTRODUCTION

This document is intended to describe the Sycon Multi drop protocol. This protocol should be common amongst all Sycon products that can be connected on a "network". The network is a network of Sycon products. This will allow an integrator to connect any Sycon product to a 2-wire 485 link and allow all these products to be friendly and useable on the same network. This also allows for a utility to scan a 485 network and determine the type and quantity and version of Sycon products on that network.

ELECTRICAL

Currently, the electrical interface for multi drop is 2-wire RS485. This allows for a single master, multi slave, half-duplex network of Sycon products. Some Sycon products may implement this protocol over a RS232 interface. This allows the same protocol stack software to be used for 232 or 485 networks, and makes for a seamless upgrade path from point to point to multi-drop setups.

BAUDS

All Sycon products implementing this protocol must be 9600 baud capable. Products may have other bauds, too, but for guaranteed compatibility to this standard, 9600 baud must be an option. All instruments on one network link must have the same baud rate.

STRUCTURE

This protocol is a byte-packet binary protocol. A packet begins with STX and ends with '\r' (carriage return). Each field in angle brackets (<>) is a byte, and is not optional. Fields in regular brackets ([]) are optional. Ellipses (…) mean one or more of the previous. The structure of the packet is below:
<STX><ADDR><CMD_RSP>[<DATA>…]<CKSUM1><CKSUM2><CR>

STX: Start of text character (hex 02). Multiple STX characters in a row are allowed. Similarly, any data between STX characters is ignored. A single STX character syncs the receiver up to receive a new message, purging any data collected since the last STX char or carriage return received.

ADDR: Address field. This is a one byte field. Valid ranges of a valid address are 10 hex to FE hex (16 to 254 decimal). Addresses less than 10 hex are not allowed as they may be mistaken for framing characters. An address of FF hex is reserved as it is used as an extention, to indicate another byte of address information follows, for products that have an address range higher than an address of FE hex.

CMD_RSP: Command/Response field. When packet is going from master to slave, RSPF bit is zero, and the RSP field (3 bits) is zero. When packet is going from slave to master, CMD bits are the same as in the message that was sent, but the RSP field will be non-zero (indicating actual unit response status). This allows the direction to be positively indicated, as well as the command preserved on the return reply.

D7	D6	D5	D4	D3	D2	D1	D0
CMD3	CMD2	CMD1	CMD0	RSPF	RSP2	RSP1	RSP0

CMD = Command. These are the commands the master can issue to the slave. All Sycon products must respond to commands in the range of 1-7. CMDS 1-7 are handled in the protocol, at the protocol layer. Applications are not to use commands 1-7 except to implement the protocol specification. CMD codes 8-15 are set aside for product application specific use.

CMD	Mnemonic	Meaning
0	----------	Cannot be used, byte could be mistaken for framing char.
1,2	Reserved	Reserved for future use in protocol stack.
3	Prod_id	Sycon product ID, returned as decimal string
4	Version	Request slave to return software version string.
5	Reset	Request slave to reset.
6	AckPF	Request slave to clear RSPF bit/flag.
7	PROTV	Request slave to return protocol stack version as decimal string.
8-15		Available for application use.

Once the packet is sent to the slave, the slave receives and acknowledges the packet, and sends it's response. In the CMD_RSP byte, the CMD bits are unchanged from the master, but the RSP bits are filled in according to the status of the slave. See table below.

RSP	Mnemonic	Meaning
0	----------	Response code must be non zero, to indicate direction.
1	OK	Command understood and executed.
2	Err_Inv_cmd	Illegal command (CMD code not valid).
3	Err_syntax	Syntax error. (too many bytes in data field, not enough bytes, etc).
4	Err_range	Data range error.
5	Err_inh	Inhibited
6	Err_obso	Obsolete command. No action taken, but not really an error.
7		Reserved for future protocol stack use

RSPF: This bit only has meaning when packet is going from slave to master. If this bit is 1, then the slave has been reset since the last ack power fail flag was received.

DATA: The optional data. Since this protocol is binary transparent, STX and carriage return characters are not allowed in the data field. To solve this problem, these characters must be escaped. To send a hex 02, send the protocol escape character (HEX 07) followed by '0' (hex 30). To send a hex 0D, send the protocol escape character (HEX 07) followed by '1' (hex 31). To send a hex 07, send the protocol escape character (HEX 07) followed by '2' (hex 32). Note that the protocol escape character (HEX 07) cannot be sent by itself, but can only be sent followed by a 0,1,or2. If the protocol escape character is sent by itself, or is not followed by a 0,1,or 2, then the packed it deemed invalid and ignored.

CKSUM1,2: The checksum characters for the message. This is the mod-256 checksum of Binary message data **except the STX and carriage return and BEFORE ESCAPE CHARACTER BYTE STUFFING**. The data is encoded as follows:

1) Compute the mod 256 checksum of ADDR, CMD_RSP and DATA fields, before byte stuffing with escape characters.

2) CKSUM1 character is the upper (most significant) four bits of the checksum (read as a nibble, 0-15, or right justified) plus hex 30 (ascii 0). This yields an ascii character from '0' (hex 30) to '?' (hex 3f).

3) CKSUM2 character is the lower (least significant) four bits of the checksum plus hex 30 (ascii 0). This yields an ascii character from '0' (hex 30) to '?' (hex 3f).

Note: If the checksum of the packet is invalid, the packet is deemed invalid and will be ignored.

Note: Invalid packets (bad checksum, too short, corrupt data, bad escape sequences) are to be ignored by slaves (and masters). That means no reply at all to an invalid packet.

OPTIONAL SERIALIZED PACKET MODE

The SMDP (Version 3 and above) protocol allows for a serial number in the packet, to positively associate a command from the master with the correct response. This is to detect errors in serial communications ports where packets are queued and re-sent out of order. The protocol structure is nearly identical, except:

1. Just before the checksum bytes, the master must place a "serial number" byte. This must be a value greater than 0x10 (16) in order to not be mistaken as a framing character.

2. The checksum seed must be hex 40, instead of hex 30. This is how the slave knows that it is receiving a packet with the extra SRLNO field.

The slave simply places the SRLNO byte into it's response packet. In order for this feature to be effective, the master should:

3. Generate a new SRLNO value for each packet that it sends out (modulo 255, and greater than 0x10, of course)
4. When an SMDP response packet is received, verify the SRLNO value matches the one it sent out.

Therefore, the packet structure to take advantage of this feature is as follows:
<STX><ADDR><CMD_RSP>[<DATA>…]**<SRLNO>**<CKSUM1><CKSUM2><CR>

<end of document>