A method and circuit for receiving frame messages connected to one or more nodes, each controlled by a microcontroller, along a common transmission line of a multiplex communication system. The method includes the steps of: (a) receiving the type and target bytes, and responsive thereto generating a first interrupt signal to the microcontroller at the node; (b) commanding the microcontroller, responsive to receiving the first interrupt signal, for qualifying the frame message by retrieving and matching one or more of the type and target bytes with corresponding bytes stored in programmable memory within the microcontroller, and responsive thereto generating a qualified signal; and (c) receiving data bytes from the frame message only responsive to receiving the qualified signal from the microcontroller.
### FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Albania</td>
<td>LS</td>
<td>Lesotho</td>
</tr>
<tr>
<td>AM</td>
<td>Armenia</td>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>LV</td>
<td>Latvia</td>
</tr>
<tr>
<td>AZ</td>
<td>Azerbaijan</td>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>BA</td>
<td>Bosnia and Herzegovina</td>
<td>MD</td>
<td>Republic of Moldova</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>MK</td>
<td>The former Yugoslav</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>MN</td>
<td>Mongolia</td>
</tr>
<tr>
<td>BI</td>
<td>Benin</td>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>MX</td>
<td>Mexico</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>CU</td>
<td>Cuba</td>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>SA</td>
<td>Saudi Arabia</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
<td>SG</td>
<td>Singapore</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>GB</td>
<td>United Kingdom</td>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>GE</td>
<td>Georgia</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>GH</td>
<td>Ghana</td>
<td>TJ</td>
<td>Tadjikistan</td>
</tr>
<tr>
<td>GN</td>
<td>Guinea</td>
<td>TM</td>
<td>Turkmenistan</td>
</tr>
<tr>
<td>GR</td>
<td>Greece</td>
<td>TR</td>
<td>Turkey</td>
</tr>
<tr>
<td>HU</td>
<td>Hungary</td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>IE</td>
<td>Ireland</td>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td>IL</td>
<td>Israel</td>
<td>Ug</td>
<td>Uganda</td>
</tr>
<tr>
<td>IS</td>
<td>Iceland</td>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>IT</td>
<td>Italy</td>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>JP</td>
<td>Japan</td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
<tr>
<td>KE</td>
<td>Kenya</td>
<td>YU</td>
<td>Yugoslavia</td>
</tr>
<tr>
<td>KG</td>
<td>Kyrgyzstan</td>
<td>ZW</td>
<td>Zimbabwe</td>
</tr>
<tr>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MULTIPLEX COMMUNICATION INTERFACE
CIRCUIT AND METHOD

This invention relates to a multiplex transmission system of the type used in automotive applications for transmitting serial data message formats among a plurality of multiplex nodes connected to a common multiplex transmission line.

Messages transmitted from an originating node in a multiplex communication system normally include priority/type bytes describing the group or function of the nodes that should receive the message, in addition to target bytes that may define the specific address or addresses nodes to receive the message. The transmitting node must receive confirmation or acknowledgment that the addressed node has received the message.

A typical received node includes interface circuitry that includes programmable memory for storing the address of the node as well as the priority and/or type functions for which the node is responsible. These stored type and target bytes must then be compared by the interface circuitry with the type and target bytes of all incoming messages in order to determine if the message should be received and processed. Since programmable memory within the interface circuit represents an added cost that increases complexity without increasing functionality, it is desirable to provide a node interface circuit that offloads this function to the host microprocessor at the node that already includes programmable memory and comparators capable of performing this function.

It is therefore an object of the present invention to provide interface circuitry that commands the host microcontroller to compare previously stored values corresponding to message identifiers that are addressed for receipt and processing at the node, so that the interface circuitry does not have to perform the storage and
comparison functions necessary to qualify, accept and acknowledge incoming messages.

The present invention includes a method and circuit for receiving frame messages connected to one or more nodes, each controlled by a microcontroller, along a common transmission line of a multiplex communication system. The method includes the steps of: (a) receiving the type and target bytes, and responsive thereto generating a first interrupt signal to the microcontroller at the node; (b) commanding the microcontroller, responsive to receiving the first interrupt signal, for qualifying the frame message by retrieving and matching one or more of the type and target bytes with corresponding bytes stored in memory within the microcontroller, and responsive thereto generating a qualified signal; and (c) receiving data bytes from the frame message only responsive to receiving the qualified signal from the microcontroller.

The invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of a node interface circuit in accordance with the present invention;

Figure 2 is a schematic block diagram of the node interface circuit and the host microcontroller that cooperate at the node of the multiplex communication system;

Figure 3 illustrates the frame message bytes and the interrupt and chip select signals processed within the interface circuit for the case where the message is not directed for the node processing the signal;

Figure 4 illustrates the frame message bytes and the interrupt and chip select signals processed within the interface circuit for the case where the host microcontroller has received and examined the priority/type and target bytes and has determined that the
frame message should be processed at the node, and has
generated an acknowledgment to the originating node;

Figure 5 illustrates the frame message bytes and the
interrupt and chip select signals processed within the
interface circuit for the case where the host
microcontroller has commanded a message transmission, and
has received and examined the transmitted priority/type
and target bytes to determine whether receipt and
acknowledgment of the message will be required if
 arbitration is lost; and

Figure 6 illustrates a block diagram of the steps in
accordance with the method of the present invention.

A circuit diagram for a typical node in a multiplex
communication system is illustrated generally in Fig. 2. and
includes a host microprocessor 10, typically a Motorola
68HC11 or a Texas Instrument TMS370, which is coupled
through a standard 5 line SPI interface 12 to the interface
integrated circuit, hereinafter referenced as the network
interface circuit 20 in accordance with the present
invention. In the first preferred embodiment, the network
interface circuit 20 is Ford part number N7100070FTCFCA that
includes host interface digital circuitry, transceiver
digital logic circuitry and analogue receive comparators,
together with miscellaneous oscillators and buffer
circuitry.

A 4 megahertz resonator 22 is coupled to the network
interface integrated circuit 20 in order to provide a stable
frequency standard. The network interface circuit 20
includes a pair of receive input signal lines 24, and a pair
of transmit signal output lines 26 that are connected to
transmit driver circuitry 30. The transmit driver circuitry
30 generates the appropriate voltage levels required to
interface with the main communications network bus 40, which
in the preferred embodiment of the present invention
operates in accordance with SAE standard J1850 (that will be
incorporated herein by reference). Receiver filter elements
28, comprising series resistors and parallel capacitors, provide input protection and noise filtering functions on the receive input PWM signals from the network bus 40.

With reference to Fig. 1, a schematic block diagram for the network interface circuit 20 is illustrated as including receive input signal lines 24 and transmit signal output lines 26 as previously described. The input lines 24 are coupled to a receiver analogue comparator 102 that includes two single ended comparators for comparing each of the input lines to a voltage threshold and also includes a differential comparator for comparing the difference in the potential between the two input signal lines 24. Three digital output lines 104 from receive analogue comparator 102 are coupled to an input of fault tolerant switchover logic and bit decoders 106. These logic circuits select between the input lines in order to minimise bit error rates and the effect of network faults. A single output 108 from the bit decoders 106 is coupled to the input of a serial to parallel converter 110. The serial bit rate at the input 108 is approximately 41.7 kilobits per second, while the output of the serial to parallel converter 110 includes 8 parallel bits clocked at one-eighth of the input frequency. The output signal of the serial to parallel converter 110 is coupled to an internal signal data bus 112 that feeds other elements of the network interface circuit 20.

The internal signal data bus 112 is coupled to message qualification registers 120, which store the first 2 bytes of the incoming message frame from the communication network bus 40. The outputs of the message qualification registers 122 are coupled to a host interface bus 130, which in turn is coupled to the host microcontroller 10 (not illustrated in Figure 1) through host interface control logic 140 and standard SPI interface 12.

Transceiver control logic 250 is responsible for controlling the reception and transmission of message frames on the network. During data reception, it controls the movement of data from the output of the serial to parallel
converter 110, to the message qualification registers 120, to a receive FIFO 150, and to a TX ACK FIFO 260. During message transmission, it controls the movement of data to a parallel to serial converter 190 from a TX buffer 230 and a node address register 160.

After the first two bytes of the incoming message frame are successfully received and stored in the message qualification registers 120, the transceiver control logic 250 instructs the host interface control logic 140 to generate an interrupt request to the host microcontroller 10. The host interface control logic 140 accomplishes this by pulling the interrupt request host interface line (INT) 300 from high to low. This interrupt request is referred to as the qualification interrupt request or first interrupt signal.

The host microcontroller 10 responds to this interrupt request by asserting the chip select (CS) signal 400 and commanding the host interface control logic 140 to fetch the contents of the message qualification registers 120 using a predetermined sequence of byte transfers across the SPI interface 12.

The host microcontroller 10 then compares the contents of the message qualification registers 120 with pre-determined lists of values for the type and target bytes stored in internal programmable memory, and determines whether the network interface circuit should continue the reception and acknowledgment of message frame. The host microcontroller 10 transmits the desired response to the network interface circuit 20 through a predetermined command sequence across the host interface 12.

The internal signal data bus 112 is coupled to an input of a receiver FIFO 150 that includes 8 bytes of storage for the remainder of the incoming message data. The receiver FIFO 150 is used to store data for message types that contain data intended for the host microcontroller 10. The 8 bytes of data typically include a source/transmit ID byte,
0-7 data bytes, a CRC byte and an EOD (End of Data) bit as illustrated in Fig. 3.

If the host microcontroller 10 has commanded the network interface device 20 to continue the reception and acknowledgment of the message frame in response to the qualification interrupt request, then following the receipt of the EOD network signal the transceiver control logic 250 begins transmission of the message acknowledgment byte(s), and instructs the host interface control logic 140 to generate an interrupt request to the host microcontroller 10. This interrupt request is referred to as the receive complete interrupt request or the second interrupt signal.

The host microcontroller 10 responds to the receive complete interrupt request by asserting the chip select signal and commanding the host interface control logic 140 to provide it with the contents of the receive FIFO 150. The host interface control logic 140 accomplishes this by moving data from the receive FIFO output 152 across interface bus 120 and across the SPI interface 12.

The type of message acknowledgment returned by the network interface circuit 20 depends upon the type of message frame that is received. The transceiver control logic 250 determines the appropriate acknowledgment format by examining the type of the received message, which is stored in the message qualification registers 120. Most message types are acknowledged by transmitting the contents of the node address register 160. Other message types are acknowledged by transmitting the contents of the ACK data register 170 followed by a CRC byte from the CRC generator 180.

The ACK acknowledgment bytes are processed in a converter 190 from a parallel to a serial bit stream, which is then fed to a pulse width modulated (PWM) generator 200. The output of the PWM generator 200 is coupled to transmitter drive circuitry 210, which in turn is coupled to the transmit signal output line 26 as previously discussed. The PWM generator 200 encodes the bits from the parallel to
serial converter 190 in accordance with the pulse width
modulation scheme of SAE standard J1850. Watchdog circuitry
220 limits the length of the pulse width modulation signal
and also limits the length of the data frame which can be
transmitted, both in accordance with requirements of SAE
standard J1850.

In order to transmit a data message, the host
microcontroller 10 sends a transmit command followed by the
data to be transmitted to the network interface circuit 20
through the SPI interface 12. The host interface logic 140
decodes the transmit command, and then routes the data to be
transmitted across the host interface bus 130 to transmit
buffer 230. The transmit buffer 230 comprises digital
storage for up to nine bytes of data. An output of the

transmit buffer 230 is coupled to an internal transmit data
bus 240, that is then coupled to the input of the parallel
to serial converter 190, the function of which has been
previously discussed. The transceiver controller 250
monitors the outputs from the switchover logic and bit
decoders 106, and determines when a valid transmission
opportunity exists on the communications network bus 40 at
the receiver input 24. When a valid transmission
opportunity occurs, the transceiver controller 250 moves
data from the transmit buffer 230, as well as the node
address register 160, to the parallel to serial converter
190. This data is then converted to PWM format by the PWM
generator 200 and transmitted by the transmit drivers 210
over the transmit output lines 26 to the main communications
network bus 40.

After completion of the transmission of the data from
the transmit buffer 230, the transceiver controller 250
commands a CRC checker/generator 180 to transfer the
contents of the generated CRC byte over the internal
transmit data bus 240 to the parallel to serial converter
190. The CRC byte is then transmitted over the transmit
output lines 26.
Following the transmission of the CRC byte, the transceiver controller 250 waits for the end of the data field, and then any acknowledgment bytes that appear on the network at the receiver inputs 24 are received and stored in a seven byte transmit acknowledgment FIFO 260. When the End of Message byte is detected, the transceiver control logic 250 instructs the host interface control logic 140 to generate an interrupt request to the host microcontroller 10 in the manner previously described. This interrupt request is referred to as the transmit complete interrupt request or third interrupt signal.

The host microcontroller 10 responds to the interrupt request by asserting the chip select line 400 and commanding the host interface control logic 140 to fetch the transmission completion status, and optionally the contents of the transmit acknowledgment FIFO, using a predetermined sequence of byte transfers across the SPI interface 12.

An oscillator circuit 270 is coupled to the three terminal resonator 22 (illustrated in Fig. 2), and provides a stable clock signal for driving the internal logic of the system. Various control and status registers 280 are provided for conducting housekeeping duties and detecting network faults.

Figs. 3, 4 and 5 illustrate how the interrupt request signals 300 are generated in relationship to various types of messages on the main communications network bus 40. With specific reference to Fig. 3, the message frame 280 comprises an SAE standard J1850 message frame and includes a start of frame network element 282, a priority/type byte 284, a target byte 286, a source address byte 288 and a CRC byte 290. The End of Data network element 292 follows the CRC byte 290, and in turn is followed by any required acknowledgment bytes 294.

With continuing reference to Fig. 3, the interrupt signal 300 from the host interface logic 140 remains high until the end of the target byte 286, at which time it transitions to low 310. This interrupt request is passed to
the host microcontroller 10 from the host interface logic 140 as previously explained. The chip select signal 400 and the system clock signal 500 are shown to illustrate the processing of the interrupt request by the host microcontroller 10. The host microcontroller 10 asserts the chip select signal 400 at transition 420, processes the qualification interrupt request, and then deasserts the chip select signal 400 at transition 430 following the completion of the interrupt processing.

The host microcontroller 10 must complete the processing of the qualification interrupt request transmitted at the end of the target byte 286 before the completion of the EOD byte 292. In this manner, the host microcontroller 10 must determine within the specified time if the remainder of the message frame is to be received and if the message frame is to be acknowledged. In the example illustrated in Fig. 3, the host microcontroller 10 has determined, by comparing the priority/type and target bytes of the message frame as stored in the qualification registers 120, that the remainder of the message frame is not to be received and an acknowledgment signal will not be transmitted. The host microcontroller 10 also has commanded the network interface circuit 20 not to generate a receive complete interrupt request following the EOD byte and not to acknowledge the message. The acknowledgment byte 294 shown in Fig. 3 is not transmitted by the network interface circuit in this example. It is used to illustrate that some other network node may acknowledge a message that is not acknowledged by this network interface circuit. The network interface circuit 20 returns the interrupt signal 300 to high at transition 330 after the host microcontroller 10 has responded to the interrupt request. The interrupt signal 300 remains high until the next interrupt request in the next message frame.

Fig. 4 illustrates the case where the host microcontroller 10 has received and examined the priority/type bytes 284 and target bytes 286, and thereby
has determined that the frame message should be processed at the node.

At transition 310 the network interface device asserts the interrupt line to notify the microcontroller 10 that the type and target of the message are available. The microcontroller 10 responds to the interrupt request by asserting the chip select signal line 400 at transition 415, and reading the type and target data bytes 284 and 286 from the message qualification registers 120. On examining these bytes and determining that the messages are intended for processing at the node, the microcontroller 10 initiates an acknowledge command and deasserts chip select signal 400 at transition 420 to complete the message qualification processing.

Following the reception of the End of Data bit 292 in the message frame, the network interface device 20 begins transmission of the message acknowledgment byte 294 and reasserts interrupt line 300 at transition 325 to notify the microcontroller 10 that the remainder of the message frame is available in the receive FIFO 150. The microcontroller 10 responds to this interrupt signal by asserting chip select 400 at transition 440, reads any required data from the receive FIFO 150, and then deasserts chip select at transition 450 to notify the network interface device 20 that the processing is complete.

Fig. 5 illustrates the process whereby a message transmission is initiated. The microcontroller 10 must first transfer the data to be transmitted to the network interface device 20 by asserting chip select signal 400 at transition 460. The transmit command is then followed by the data to be transmitted over the SPI lines 12 from the microcontroller 10 to the host interface logic 140. The end of the transmit command is designated by the deassertion of the chip select signal 400 at transition 470. The network interface device 20 then waits for 3 idle bit periods on the main communications network bus 40 or for a start of frame signal to be detected, at which time the microcontroller 10
begins the transmission of the data stored in the transmit buffer 230 in the manner previously discussed.

Note that the interrupt signal 300 is asserted at transition 310 following the reception of the priority/type and target bytes, 284 and 286 respectively. The microcontroller 10 must then perform the message qualification function as previously described. All messages that are transmitted must also go through the message qualification procedure to protect against the possibility that arbitration will be lost on the network requiring the remainder of the message frame to be processed as a reception. When arbitration is lost, the network interface device 20 changes from being a transmitter to a receiver of the message frame, and must determine if the message frame is to be received or ignored. When transmission of the message frame has been completed and the node receiving the message has acknowledged receipt, another interrupt request is generated at transition 370 to notify the host microcontroller 10 that a transmission of the frame has been completed.

A flow chart for the process described above is illustrated with reference to Fig. 6. At step 600 the network interface device 20 receives the priority/type 284 and target 286 bytes and stores these signals in qualification registers 120. At step 620 the network interface circuit 20 generates an interrupt request to the microcontroller 10 in response to receipt of the priority/type and target bytes. In response to this interrupt request, at step 630 the microcontroller 10 examines the qualification registers 120 to determine if the message is directed toward the node by comparing the contents of the registers to data stored in tables within the microcontroller 10. At the same time in step 640 the network interface device 20 continues the reception and storage of the message frame, specifically the source and data bytes, in the receive FIFO 150.
At step 650 the microcontroller 10 transmits the qualification command to the host interface circuit 20 indicating that the message is either qualified or not qualified. If the message is not qualified, processing of the message frame terminates and the process moves to step 700 which requires waiting for the next message frame. If the message is qualified, then the process moves to step 660 wherein the loading of the received message into the receive FIFO 150 continues until the EOD byte is received. At step 664, if the message data bytes are received through EOD without error, then the process moves to steps 688 and 690. Otherwise the process moves to step 700. At step 668, if the message is of a type that contains data to be received by the host, then the process moves to step 670. Otherwise, the process moves to step 682. At step 670, the network interface circuit 20 generates a receive complete interrupt request to the host microcontroller 10. At step 680 the microcontroller 10 responds to the receive complete interrupt request and retrieves the contents of the receive FIFO 150 and continues processing the received data. At step 682 any required host processing of any received data is complete. In parallel with steps 668, 670, 680 and 682, the network interface device 20 transmits the acknowledgment byte(s) to the transmitting node at step 690 indicating that the message has been received and is being processed. Following step 690 the network interface device 20 waits for the next message at step 700.

The contents of the qualification registers 120 are transmitted to the host microcontroller 10 for comparison with previously stored values corresponding to message identifiers that are addressed for receipt and processing at a node. In this manner the network interface device 20 does not need to program, store and compare the priority/type and target bytes in the process of determining if the message is intended for receipt and processing at the node. These functions, which require the use of more expensive programmable memory storage devices, may be accomplished in
a more economical and efficient manner within the already existing resources of the microcontroller 10.
CLAIMS

1. A method for receiving frame messages communicated to one or more nodes, each controlled by microcontroller, along a common transmission line of a multiplex communications system, comprising the steps of:
   (a) receiving type and target bytes and responsive thereto generating a first interrupt signal to the microcontroller,
   (b) commanding the microcontroller, responsive to receiving the first interrupt signal, for qualifying the frame message by retrieving and matching one or more of the type and target bytes with corresponding bytes stored therein, and responsive thereto generating a qualified signal,
   (c) receiving data bytes from the frame message only responsive to receiving the qualified signal from the microcontroller.

2. A method as claimed in claim 1, wherein step (b) further includes the substep:
   (b1) generating the qualified signal only responsive to one of the type and target bytes matching corresponding bytes stored in programmable memory within the microcontroller, whereby the qualified signal is representative of the frame message being directed to the node for processing.

3. A method as claimed in claim 1 or 2, wherein step (c) further includes the substep:
   (c1) loading, only responsive to receiving said qualified signal, data bytes from the frame message into temporary memory for subsequent processing.

4. A method as claimed in any one of claims 1 to 3, further comprising the steps of:
(d) calculating a CRC value form, and generating a corresponding CRC signal representative of, the integrity of the data bytes received in the frame message; and

(e) generating a second interrupt signal responsive to receiving a valid CRC signal for commanding the microcontroller to fetch the data bytes from the temporary memory.

5. An interface device for receiving frame messages communicated to one or more nodes, each controlled by a microcontroller, along a common transmission line of a multiplex communications system, comprising in combination:

a qualification memory for receiving type and target bytes of the frame message, and responsive thereto generating a first interrupt signal;

an interface controller, coupled between said qualification memory and the microcontroller, for commanding, responsive to said first interrupt signal, the microcontroller to retrieve and match one or more of the type and target bytes with corresponding bytes stored in the microcontroller, and responsive thereto generating a qualified signal; and

a receive memory for receiving and storing responsive to receiving said qualified signal, data bytes of the frame message, whereby the interface controller and not the microcontroller qualifies the type and target bytes to enable the receiving and storage of the data bytes.

6. An interface device as claimed in claim 6, wherein the microcontroller includes programmable memory for storing therein selected type and target bytes representative of the function(s) and address of the node, and means for comparing the type and target bytes from the frame message therewith, and wherein said interface controller includes means for generating said qualified signal responsive only to a match identified by the microcontroller, whereby the
microcontroller and not said interface controller qualifies the frame message as intended for the node.

7. An interface device as claimed in claim 5 or 6, further including temporary memory, coupled to said interface controller and the transmission line of the multiplex communications system, for receiving and storing therein data bytes from the frame message pending receipt of said qualified signal.

8. An interface device as claimed in claim 7, further comprising CRC means, coupled to the transmission line of the multiplex communication system, for receiving the frame message and generating a CRC signal representative of the integrity of the data bytes.

9. An interface device as claimed in claim 8, further including means for generating an acknowledge signal responsive to receiving both a valid CRC signal and an End of Data byte, whereby the node originating the message frame will receive acknowledgment that the frame message has been received by the intended node(s).

10. An interface device as claimed in claim 9, wherein said means for generating said acknowledge signal comprises:

    means for commanding the microcontroller, responsive to receiving said valid CRC signal, to fetch the data bytes from said temporary memory;

    and means for generating said acknowledge signal responsive to receiving said valid CRC signal and an End of Data byte from the frame message.
FIG. 5
RECEIVE PRI/TYPE & TARGET BYTES
STORE IN QUAL REGISTER

GENERATE INTERRUPT REQUEST

MICRO COMPARES QUAL
REGS TO TABLE TO ACCEPT
OR REJECT MESSAGE

STORE SOURCE & DATA INFO IN
RECEIVE FIFO

MICRO SENDS QUAL
COMMAND TO HOST MICRO INTERFACE

LOAD RECEIVE FIFO AND
CONTINUE TO END OF MESSAGE

MESSAGE CRC VALID

YES

TRANSMIT MESSAGE ACK
TO ORIGINATING NODE

WAIT FOR NEXT MESSAGE ON BUS

FIG. 6
**INTERNATIONAL SEARCH REPORT**

**A. CLASSIFICATION OF SUBJECT MATTER**

| IPC 6 | H04L29/06 |

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

| IPC 6 | H04L |

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>WO 84 03192 A (AT&amp;T COMPANY) 16 August 1984 see page 3, line 24 - page 4, line 13 see page 5, line 1 - line 11 see page 9, line 27 - line 30 see page 12, line 28 - page 14, line 37 see page 17, line 14 - line 31 see page 31, line 14 - line 23 ---</td>
<td>1, 3, 5, 9</td>
</tr>
<tr>
<td>Y</td>
<td>US 5 319 752 A (B. PETERSSEN ET AL.) 7 June 1994 see abstract see column 2, line 55 - line 64 see column 3, line 6 - line 26 ---</td>
<td>4, 10</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of box C.

**Patent family members are listed in annex.**

* Special categories of cited documents:
  * A: document defining the general state of the art which is not considered to be of particular relevance
  * E: earlier document but published on or after the international filing date
  * L: document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  * O: document referring to an oral disclosure, use, exhibition or other means
  * P: document published prior to the international filing date but later than the priority date claimed

**Date of the actual completion of the international search**

22 August 1997

**Name and mailing address of the ISA**

European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tdl. (+31-70) 340-2040, Tx. 31 651 epo nl Fax (+31-70) 340-3016

**Date of mailing of the international search report**

05.09.97

Authorized officer

Larincese, C
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 5 485 584 A (R. HAUSMAN ET AL) 16</td>
<td>1-10</td>
</tr>
<tr>
<td></td>
<td>January 1996</td>
<td></td>
</tr>
<tr>
<td></td>
<td>see column 1, line 41 - line 53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>see column 3, line 16 - line 33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>see column 7, line 21 - line 53</td>
<td></td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>WO 8403192 A</td>
<td>16-08-84</td>
<td>CA 1223326 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 0137804 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 4654654 A</td>
</tr>
<tr>
<td>US 5319752 A</td>
<td>07-06-94</td>
<td>AU 674341 B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU 4930093 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 2143951 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 0660955 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2584957 B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 7507173 T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 9407201 A</td>
</tr>
<tr>
<td>US 5485584 A</td>
<td>16-01-96</td>
<td>US 5412782 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 0577115 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 6168196 A</td>
</tr>
</tbody>
</table>