

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
20 September 2001 (20.09.2001)

PCT

(10) International Publication Number
WO 01/69408 A1

(51) International Patent Classification⁷: **G06F 15/173**

(21) International Application Number: PCT/US01/40323

(22) International Filing Date: 16 March 2001 (16.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/189,919 16 March 2000 (16.03.2000) US
Not furnished 15 March 2001 (15.03.2001) US

(71) Applicant: **WORLD WIRELESS COMMUNICATIONS, INC.** [US/US]; Suite 340, 5670 Greenwood Plaza Boulevard, Greenwood Village, CO 80111 (US).

(72) Inventors: **DAVIS, Michael, Madison**; 2006 W. 48th Street, Westwood, KS 66205 (US). **DOWNEY, David, Joseph**; 3225 West 203 Street, Louisberg, KS 66053 (US). **HOLLCROFT, Keith, Douglas**; 12721 Benson,

Overland Park, KS 66213 (US). **SHERWOOD, Charles, Gregory**; 10975 Gardner Road, Olathe, KS 66061 (US). **TEUSCHER, Ted, Andrew**; 9109 Lichtenauer #294, Lenexa, KS 66219 (US). **WALLACE, Donald, Ian**; 4412 Orofino Place, Castle Rock, CO 80104 (US).

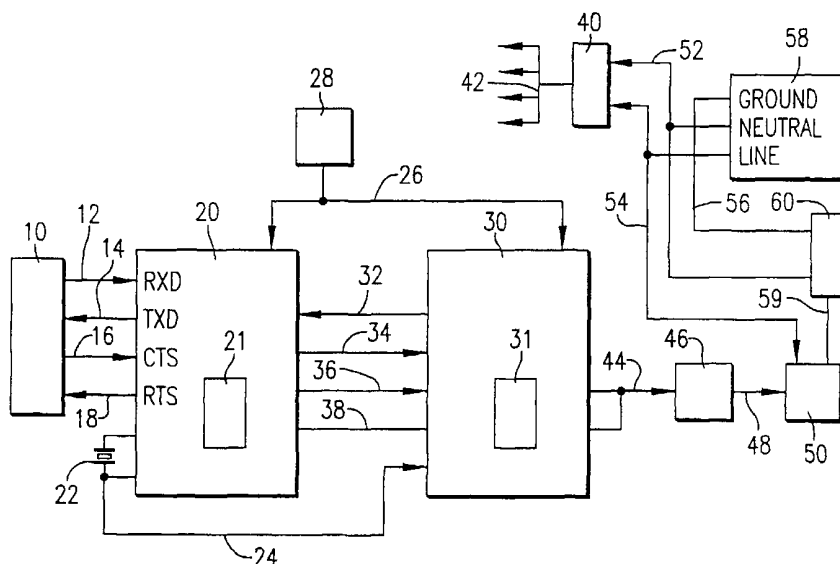
(74) Agent: **LUEBBERING, Thomas, B.**; Hovey, Williams, Timmons & Collins, Suite 400, 2405 Grand Boulevard, Kansas City, MO 64108 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: AN IMPROVED METHOD OF CONNECTING DEVICES TO THE INTERNET



(57) Abstract: A system which performs the function of monitoring and/or controlling a device, communicates with the Internet using Internet protocols, and is able to serve a web page or communicate with a server over the Internet to allow remote control of the device. The system can include an Internet node comprising a microcontroller (20) programmed to communicate with the Internet and serve a web page to a browser. A second microcontroller (30), is programmed to control an appliance and the two microcontrollers (20, 30) communicate with each other on an internal bus (38). An Internet gate is provided to control connections between one or more Internet nodes, serving one or more appliances, and the Internet. The Internet gate includes a microcontroller to control communications between a plurality of Internet nodes and the Internet gate and between the Internet gate and the Internet.



WO 01/69408 A1



Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

1 AN IMPROVED METHOD OF CONNECTING DEVICES TO THE INTERNET

3 FIELD OF THE INVENTION

4 The present invention relates to improvements in the field of Internet connectivity.
5 More particularly, the invention relates to a small area low cost system for connecting
6 devices such as lamps, coffeepots, thermostats, home security systems or other devices to the
7 Internet.

9 BACKGROUND

10 Large publicly or privately accessible computer networks, such as the Internet,
11 provide infrastructure for establishing remote access connections via computing devices for
12 the control of electrical and electronic devices. Standardized communication protocols are
13 used on these networks that may be employed to remotely retrieve status information and/or
14 send control information to addresses anywhere on the network. In the case of the Internet,
15 as is well know to one of ordinary skill, this communication protocol is known as TCP/IP.
16 The TCP/IP protocol allows a computing device connected to the Internet to specify a
17 particular Internet address and connect to the computing device which was given that unique
18 address. By using standard communication commands and responses developed under the
19 TCP/IP protocol, signals can be sent across the network and between any computing
20 platforms that support the protocol. Included within the TCP/IP protocol are commands that
21 may be sent by a web browser to a given address to retrieve information or to send
22 information. Thus, web browsers may be used to retrieve and send information to devices
23 that monitor and control remote electronic devices.

24 Traditionally, relatively large and expensive computer systems, such as personal
25 computers (PCs), have been required to make connections to the Internet. This makes it
26 relatively expensive to remotely connect to and control a device, such as a thermostat,
27 through the Internet. Prior art home control devices, such as X-10 Active Home
28 manufactured by X-10 USA, Inc. are able to control home appliances through the Internet.
29 Such systems provide small modules used to connect home appliances to a central controller.
30 However, these systems still require a personal computer to connect between the in home
31 system and the Internet. Internet remote control of appliances of many varieties would be far
32 more efficient if relatively inexpensive means for connecting those appliances to the Internet
33 were available. Such connections would allow control by homeowners remotely while using
34 a web browser. Moreover, internet access and control would be useful for other types of

1 devices such as vending machines wherein account information may be checked, debited, or
2 credited via an internet connection.

3 A need has therefore arisen for a small, low cost, and efficient method to remotely
4 control an appliance through the Internet without the need for a PC to have access to the
5 Internet.

6 It is desirable then to provide a low cost and relatively small area Internet connection
7 access device which may be used with existing electronically controllable devices such as
8 household appliances, vending machines, or other appliances to allow monitoring and control
9 of these devices through the Internet.

10

11 SUMMARY OF THE INVENTION

12 The present invention solves the above-described problems and provides a distinct
13 advance in the art of remote electronic control of devices. More particularly, the present
14 invention provides a method and apparatus for providing an Internet access device that allows
15 the connection of appliances and other controllable devices to the Internet, and interfaces
16 with the Internet using existing protocols to allow control of the appliances via a web page.

17 The invention broadly includes a system which performs the function of monitoring
18 and/or controlling a device, communicates with the Internet using Internet protocols, and is
19 able to serve a web page or communicate with a server over the Internet to allow remote
20 control of device. In one separate aspect of the invention, the system can include an Internet
21 node comprising a microcontroller programmed to communicate with the Internet and serve a
22 web page to a browser. A second microcontroller, is programmed to control an appliance and
23 the two microcontrollers communicate with each other on an internal bus. In another
24 embodiment, the same microcontroller performs both of the above described functions. The
25 web page allows a user to monitor or control an application from a remote location via access
26 to the web page address.

27 In a second separate aspect of the invention an Internet gate is provided to control
28 connections between one or more Internet nodes, serving one or more appliances, and the
29 Internet. The Internet gate includes a microcontroller to control communications between a
30 plurality of Internet nodes and the Internet gate and between the Internet gate and the
31 Internet. A UART, under microcontroller control, provides communication capability
32 between multiple Internet nodes and the Internet. Connection to the Internet is made via a
33 10/100 BT connection or some other connection type such as a modem. The Internet gate is

1 capable of serving multiple Internet nodes by responding to synchronization, finish, and echo
2 packets and sending all other packets on to the attached Internet nodes.

3 In a third separate aspect of the invention, an Internet node is provided with a LRS1-T
4 remote access server in the absence of an Internet gate.

5 By constructing an Internet node alone or in combination with an Internet gate,
6 common household appliances and other appliances can be connected to the Internet to allow
7 for remote control without requiring an expensive computing device such as a personal
8 computer to make the connection.

9 These and other important aspects of the present invention are described more fully in
10 the detailed description below.

11 12 BRIEF DESCRIPTION OF THE DRAWINGS

13 A preferred embodiment of the present invention is described in detail below with
14 reference to the attached drawing figures, wherein:

15 Fig. 1 is a schematic block diagram of an embodiment of an Internet node of the
16 present invention;

17 Fig. 2 is a schematic block diagram of another embodiment of an Internet node of the
18 present invention;

19 Fig. 3 is a schematic block diagram of another embodiment of an Internet node of the
20 present invention;

21 Fig. 4 is a flow diagram of an embodiment of a computer program of the present
22 invention;

23 Fig. 5 is a flow diagram of an embodiment of a computer program of the present
24 invention;

25 Fig. 6 is a flow diagram of an embodiment of a computer program of the present
26 invention; and

27 Fig. 7 is a schematic block diagram of an embodiment of an Internet gate of the
28 present invention.

29 The drawing figures do not limit the present invention to the specific embodiments
30 disclosed and described herein. The drawings are not necessarily to scale, emphasis instead
31 being placed upon clearly illustrating the principles of the invention.

1 DETAILED DESCRIPTION

2 Reference will now be made to the drawings in which the various elements of the
3 present invention will be given numeral designations and in which the invention will be
4 discussed so as to enable one skilled in the art to make and use the invention. It is to be
5 understood that the following description is only exemplary of the principles of the present
6 invention, and should not be viewed as narrowing the claims.

7 Turning now to Fig. 1, a home appliance Internet node constructed in accordance with
8 a preferred embodiment of the invention is illustrated. The Internet node is operable for
9 connecting a device, such as a home appliance, to the Internet. The Internet node broadly
10 includes two microcontrollers, each with a program memory store containing software that
11 programs one of the microcontrollers to communicate with the appliance and the other
12 microcontroller to serve a web page or communicate with a server over the Internet.

13 The Internet node comprises a connector 10 that connects to microcontroller 20
14 through four lines. The lines are Receive Data (RXD) 12, Transmit Data (TXD) 14, Clear to
15 Send (CTS) 16, and Request to Send (RTS) 18. These lines form a serial interface between
16 microcontroller 20 and a serial device (not shown) connected to connector 10. Connector 10
17 serves as a connection between the Internet node and an Internet gate or other device that
18 provides a physical connection to the Internet. An alternative embodiment would use a
19 digital data radio connected to connector 10 and another digital data radio that connects to an
20 Internet gate.

21 Microcontroller 20 communicates to the Internet over lines 12, 14, 16, 18 and
22 connector 10. RAM/ROM program and working memory 21 reside within microcontroller
23 20 and contain the software code and working memory for communicating with the Internet
24 and for communicating with application microcontroller 30. Microcontroller 30 also contains
25 RAM/ROM program and working memory 31. This memory holds the code used to
26 communicate with microcontroller 20 and the code that controls or monitors the appliance, in
27 this example a solid-state relay. Crystal 22 provides the master clock for microcontroller 20
28 and microcontroller 30. Microcontroller 20 contains an internal oscillator and crystal 22 sets
29 the reference frequency of the oscillator. Line 24 carries the reference frequency to
30 microcontroller 30. Both microcontrollers 20 and 30 are commercially available from Atmel
31 Corporation under the designation AT90S5815-8AC. An alternative embodiment would
32 combine the functions of both microcontrollers into one microcontroller.

33 Bus 38 is an 8-bit data bus used for communication between microcontroller 20 and
34 30. Line 36 is a write/read line used to control the direction of data flow across the bus 38.

1 When line 36 is high, data flows from microcontroller 20 to 30. When line 36 is low, data
2 flows from microcontroller 30 to 20. Line 32 is a clock line used to clock data that is being
3 sent from microcontroller 30 to microcontroller 20. Line 34 is a clock line used to clock data
4 being sent from microcontroller 20 to microcontroller 30.

5 Power supply 40 creates a regulated DC voltage sufficient to operate the various
6 electronic components depicted in Fig. 1. Lines 42 distribute the regulated voltage to the
7 various components. Reset device 28 monitors the voltage on line 42 and generates a reset
8 signal that is sent to both microcontrollers on line 26 if the voltage is not within the correct
9 tolerance. The reset device is commercially available from Motorola under the designation
10 MC34064D-5.

11 Power supply 40 receives AC voltage from input connector 58 and output connector
12 60, which are controlled by triac 50. Input connector 58 is connected to power supply 40
13 through AC neutral line 52 and AC line 54. Input connector 58 is connected to output
14 connector 60 through ground line 56 and AC neutral line 52. Triac 50 connects AC line 54 to
15 output connector 60 through line 59. Triac 50 turns power on or off to output connector 60.
16 Triac 50 is controlled by zero crossing detector 46 via line 48. Zero crossing detector 46 is
17 controlled by microcontroller 30 through line 44, which is driven by two output pins of
18 microcontroller 30 so that sufficient current drive is available to detector 46. Zero crossing
19 detector 46 only turns triac 50 on or off with the AC line is crossing through zero volts. This
20 prevents large current flows that would occur if triac 50 were activated when the AC voltage
21 was significantly different than zero. Both zero crossing detector 46 and Triac 50 are
22 commercially available from many sources under the generic designation MOC3043 and
23 MAC210A respectively.

24 Referring now to Fig. 2, there is shown a block diagram of another embodiment of the
25 Internet node of the present invention. The Internet node of Fig. 2 provides a serial interface
26 that may be used to connect many different devices to the Internet. Devices with an RS-232
27 interface or a RS-485 interface can be connected to this Internet node.

28 The Internet node of Fig. 2 is comprised of connector 110 that connects to
29 microcontroller 120 through four lines. The lines are Receive Data (RXD) 112, Transmit
30 Data (TXD) 114, Clear to Send (CTS) 116, and Request to Send (RTS) 118. These lines
31 form a serial interface between microcontroller 120 and a serial device (not shown)
32 connected to connector 110. A RS-485 interface to microcontroller 120 is also supported by
33 connector 182 and RS-485 transceiver 180. Lines 188 and 190 are the differential RS-485
34 interface lines and connector 182 provides for a connection between these lines and an

1 external device. Line 184 enables the RS-485 receiver and line 186 enables the RS 485
2 transmitter. Connector 110 or connector 182 serves as a connection between the Internet
3 node and an Internet gate or other device that provides a physical connection to the Internet.
4 An alternative embodiment would use a digital data radio connected to connector 110 and
5 another digital data radio that connects to an Internet gate.

6 Microcontroller 120 communicates to the Internet over lines 112, 114, 116, 118 and
7 connector 110. An alternative connection would use the RS-485 interface supported by
8 connector 182 and transceiver 180. Microcontroller 120 includes RAM/ROM program and
9 working memory 121, which contain the software code and working memory for
10 communicating with the Internet and for communicating with application microcontroller
11 130. Microcontroller 130 also contains RAM/ROM program and working memory 131.
12 This memory holds the code used to communicate with microcontroller 120 and the code that
13 controls or monitors the device to be connected to the Internet. In this case, by changing the
14 code contained within RAM/ROM program and working memory 131, many different
15 devices and applications can be supported. Crystal 122 provides the master clock for
16 microcontroller 120 and microcontroller 130. Microcontroller 120 contains an internal
17 oscillator and crystal 122 sets the reference frequency of the oscillator. Line 124 carries the
18 reference frequency to microcontroller 130. Both microcontrollers 120 and 130 are
19 commercially available from Atmel Corporation under the designation AT90S8515-8AC. An
20 alternative embodiment would combine the functions of both microcontrollers into one
21 microcontroller.

22 Bus 138 is an 8-bit data bus used for communication between microcontroller 120 and
23 130. Line 136 is a write/read line used to control the direction of data flow across the bus
24 138. When line 136 is high data flows from microcontroller 120 to 130 and when line 136 is
25 low data flows from microcontroller 132 to 120. Line 134 is a clock line used to clock data
26 that is being sent from microcontroller 120 into microcontroller 130. Line 132 is a clock line
27 used clock data that is being sent from microcontroller 130 into microcontroller 120.

28 Power supply 140 creates a regulated DC voltage from a higher unregulated voltage,
29 the regulated voltage is sufficient to operate the various electronic components depicted in
30 Fig. 2. Lines 142 distribute the regulated voltage to the various components. Reset device
31 128 monitors the voltage on lines 142 and generates a reset signal that is sent to both
32 microcontrollers on line 126 if the voltage is not within the correct tolerance. The reset
33 device is commercially available from Motorola under the designation MC34064D-5.

1 Application microcontroller 130 can monitor various appliances using a RS-232 or
2 RS-485 interface. Lines RXD 150, TXD 152, CTS 154, and RTS 156 connect the RS-232
3 lines from microcontroller 130 to RS-232 level translator 160. Translator 160 shifts voltages
4 between TTL levels output by microcontroller 130 on the RS-232 lines to RS-232 levels that
5 are carried by lines 168, 170, 172, and 174. Those skilled in the art will realize that for RS-
6 232 voltage compatibility a level converter is required to convert between TTL levels and
7 RS-232 levels. Translator 160 is commercially available from Maxim under the designation
8 MAX232CSE. Lines 168, 170, 172, and 174 carry signals between translator 160 and
9 connector 164 which may be connected to an appliance. A second option is provided for a
10 RS-485 interface. If the RS-485 interface will be used then the RS-232 interface translator
11 160 is not installed. If translator 160 were installed and an attempt was made to use the RS-
12 485 transceiver 162 then the output driver of translator 160 and RS-485 transceiver 162
13 would both drive the RXD line 150 causing a conflict and the inability of microcontroller 130
14 to correctly receive data. Both RS-485 transceivers 180 and 162 are commercially available
15 from Linear Technology under the designation LTC1483CS8. RS-485 transceiver 180 is
16 controlled by microcontroller 120 using receive enable line 184 and transmit enable line 186.
17 When the receive enable line 184 is low the receiver is enabled and when the transmit enable
18 line 186 is high the transmitter is enabled. Similarly, RS-485 transceiver 162 is controlled by
19 microcontroller 130 using receive enable line 158 and transmit enable line 160. When the
20 receive enable line 158 is low the RS-485 receiver is enabled and when the transmit enable
21 line 161 is high the RS-485 transmitter is enabled. Connector 182 and connector 166 allow
22 for connection of a wire bus to the RS 485 transceivers. Positive data is carried by line 188
23 between transceiver 180 to connector 182, negative data is carried by line 190 between
24 transceiver 180 and connector 182. Similarly, positive data is carried by line 176 between
25 transceiver 162 and connector 166, negative data is carried by line 170 between transceiver
26 162 and connector 166. The ground reference is provided at both connectors 166 and 182 by
27 lines 179.

28 Fig. 3 is a block diagram of another embodiment of the Internet node of the present
29 invention. This Internet node can be used to connect a vending machine to the Internet for
30 the purpose of providing a cashless and cardless method of purchasing products from the
31 machine.

32 The Internet node of Fig. 3 comprises a connector 210 that connects to
33 microcontroller 220 through four lines. The lines are Receive Data (RXD) 212, Transmit
34 Data (TXD) 214, Clear to Send (CTS) 216, and Request to Send (RTS) 218. These lines

1 form a serial interface between microcontroller 220 and a serial device connected to
2 connector 210. Connector 210 serves as a connection between the Internet node and a digital
3 data radio (not shown). A second digital data radio is connected to an Internet gate. An
4 alternative embodiment would comprise a RS-232 level converter between microcontroller
5 220 and connector 210 allowing a RS-232 device to connect to the Internet node, or the
6 interface could be adapted for RS-485 transceivers as described above.

7 Microcontroller 220 communicates to the Internet over lines to 212, 214, 216, 218,
8 and connector 210. In the preferred embodiment a digital data radio is connected to
9 connector 210. Those skilled in the art will realize the other devices, such as an Internet gate,
10 could be directly connected to connector 210. RAM/ROM program networking memory 221
11 contained the software coat networking memory or communicating with the Internet and for
12 communicating with application microcontroller 230. Microcontroller 230 also contains
13 RAM/ROM program and working memory 231. This memory holds the code used to
14 communicate with microcontroller 220 and a third microcontroller 160. Memory 231 also
15 holds the code that controls and monitors a structure commonly found in vending machines
16 called a Multi Drop Bus (MDB), which is a bus that allows partial control of the machine.
17 The MDB will be further described below. The MDB is connected to via connector 264. In
18 this application, the MDB is used to indicate that funds are available for vending products
19 from the machine. The sole purpose of microcontroller 160 is to control a display, such as a
20 Vacuum Fluorescent Display (VFD). Crystal 222 provides the master clock for
21 microcontrollers 220, 230, and 160. Microcontroller 220 contains an internal oscillator and
22 crystal 222 sets the reference frequency of the oscillator. Line 224 carries the reference
23 frequency to microcontroller 230 and microcontroller 160. All three microcontrollers 220,
24 230, and 160 are commercially available from Atmel Corporation under the designation
25 AT90S8515-8AC. An alternative embodiment would combine the functions of all three
26 microcontrollers into one microcontroller.

27 Bus 238 is an eight bit data bus used for communication between microcontroller 220
28 and 230. Line 236 is a write/read line used to control the direction of data flow across the bus
29 238. When line 236 is high data flows from microcontroller 220 to 230 and when line 236 is
30 low data flows from microcontroller 230 to 220. Line 232 is a clock line used to clock data
31 being sent from microcontroller 230 into microcontroller 220. Line 234 is a clock line used
32 to clock data being sent from microcontroller 220 into microcontroller 230. Line 233 is used
33 by microcontroller 230 to signal to microcontroller 220 that a connection to the Internet
34 should be initiated.

1 Power supply 240 creates a regulated DC voltage sufficient to operate the various
2 electronic components depicted in Fig. 3. Power supply 240 gives a higher unregulated
3 voltage from the MDB connector 264 via line 243. Lines 242 to distribute the regulated
4 voltage to the various components. Reset device 228 monitors the voltage on line 242 and
5 generates a reset signal that is sent to each microcontroller on line 226 if the voltage is not
6 within the correct tolerance. The power supply control is commercially available from Linear
7 Technology under the designation LTC1076CQ-5. The reset device is commercially
8 available from Motorola under the designation MC34064D-5.

9 Line 250 allows microcontroller 230 to receive data from the MDB interface. Optical
10 isolator 260 detects current pulses on the MDB which are carried from connector 264 over
11 line 172 to the isolator. Isolator 260 converts the current pulses to a digital logic voltage
12 level for reception by microcontroller 230. The MDB has been in the prior art for many years
13 and its operation is well understood by those of ordinary skill in the art. When
14 microcontroller 230 needs to send control signals to the MDB its sends logic voltage signals
15 on line 252 to optical isolator 170. Optical isolator 170 converts the logic voltage signals into
16 current pulses which are sent over line 174 to the MDB through connector 264.
17 Microcontroller 230 enables optical isolator 170 by sending a high signal on line 166 to
18 optical isolator 170. Note that two microcontroller pins drive line 166 to provide sufficient
19 current drive for optical isolator 170. The optical isolators are commercially available from
20 NEC under the designation PS2701-2. Line 166 also connects to the input of buffer 164
21 which itself is connected to microcontroller 160 via line 162. Line 252 is connected to line
22 166 through resistor 168. When microcontroller 230 places a high signal on line 166 to
23 enable optical isolator 170 it also feeds buffer 164 driving line 162 high. Transmit data that
24 is sent by microcontroller 230 over line 252 cannot flow through resistor 168 to
25 microcontroller 160 because line 166 is high. When microcontroller 230 needs to send data
26 to microcontroller 160 line 166 is tri-stated by microcontroller 230. This simultaneously
27 removes drive current from isolator 170 and allows data to flow from microcontroller 230
28 over line 252 through resistor 168 into buffer 164 onto line 162 and into microcontroller 160.
29 Note that buffer 164 is actually two parts of a hex inverter. The hex inverter is commercially
30 available from many sources under the generic designation 74HC04.

31 The sole intent of microcontroller 160 is to take the serial data from microcontroller
32 230 and convert it into the signals necessary to control a display, such VFD, that would be
33 connected to connector 266. Bus 278 carries eight data lines to the display and bus 276
34 carries three control lines to the to display. The display interface is standard for this class of

1 display device has been in the prior art for many years. An alternative embodiment would
2 combine the functions of microcontroller 220, 230, and 160 into one microcontroller.

3 Connector 265 is used to connect a 16 key keypad to microcontroller 230 over bus
4 262. Bus 262 contains four row lines and four column lines. The rows and columns create a
5 matrix in the keypad for 16 keys. Microcontroller 230 contains internal pull up resistors on
6 the column lines. Data from the keypad is retrieved using standard keypad scanning
7 algorithms that are available in the software prior art for many years.

8 Referring now to Fig. 3 and Figs. 4a-4d, Figs. 4a-4d show a flowchart depicting the
9 process of vending a product from a vending machine that includes an Internet node of the
10 present invention. The process begins at step 1000. Step 1002 depicts the user pressing clear
11 button on the keypad of the vending machine. The application microcontroller 230 detects
12 the clear button on a keypad connected to connector 264 using the keypad scanning code
13 mentioned above.

14 In step 1004 the application microcontroller 230 sends a message over line 252 to the
15 display microcontroller 160. This is accomplished by tri-stating line 166 so that the voltage
16 level on line 252 is detected by buffer 164 and passed on line 162 to microcontroller 160.
17 Microcontroller 160 then outputs the message over data bus 278 to connector 266 with the
18 correct control signals on bus 276 such that a display will correctly display the message.
19 When line 166 is tri-stated optical isolator 170 will not pass the signals present on line 252 to
20 the MDB.

21 Step 1006 causes the application microcontroller 230 to wait for the user to enter a
22 user ID on the keypad. Once the user ID has been entered and enter button pushed, step 1008
23 causes a prompt for the user pass code to be displayed on the display using display
24 microcontroller 160 and associated interfaces and buses as described above.

25 Step 1010 waits for the user to enter the pass code and press enter. Once the pass
26 code has been entered, at step 1012 application microcontroller 230 sends a packet to the
27 communication microcontroller 220. The packet instructs the communication
28 microcontroller to seek authorization to vend. The packet is sent over bus 238 with line 236
29 set low to indicate that data is flowing from microcontroller 230 to 220. Line 232 clocks the
30 data from microcontroller 230 into microcontroller 220.

31 Communication microcontroller 220 then attempts to initiate a TCP session with a
32 remote server as depicted in step 1014. Microcontroller 220 sends a TCP packet over line
33 212 to a device connected to connector 210, usually a wireless data radio. The data radio
34 communicates with another data radio connected to an Internet gate that for words the TCP

1 packet to the Internet. The IP address for the server is stored in RAM/ROM program and
2 working memory 221.

3 At step 1016 application microcontroller 230 causes a message to be sent to the
4 display microcontroller 160 and onto the display using the method described above. The
5 contents of the message is "Please Wait."

6 Program flow proceeds to step 1018 where a test is performed by communication
7 microcontroller 220 to determine if a TCP session was initiated with the remote server. If the
8 session didn't succeed then step 1050 is performed. This tests to determine if the user pushed
9 the cancel button. If the cancel button was not pushed, then the program flow returns to 1018
10 as depicted by step 1052. If the user did push cancel then step 1054 is performed. This step
11 returns to vending machine to the idle state. The vending machine is idled by activating line
12 166 and making it high so that data can be sent to the MDB from line 252 through optical
13 isolator 170 and on to line 174 to MDB connector 264. A return to idle message is sent by
14 microcontroller 230 to the vending machine. At step 1056 a message is passed by
15 application microcontroller 230 to communication microcontroller 220 instructing the
16 communication microcontroller to close the TCP session. At step 1058 program flow returns
17 to the beginning, step 1000.

18 Returning to step 1018, if a TCP session was successfully opened program flow
19 proceeds to step 1019. Communication microcontroller 220 sends a vend session request to
20 the remote server based on a message sent by the application microcontroller 230 over bus
21 238.

22 At step 1017 communication microcontroller 220 waits for a vend session response
23 from the remote server. If the communication microcontroller does not receive a vend
24 session response and program flow moves to step 1050 to test whether the user has pushed
25 the cancel key. If the cancel key was pushed then steps 1054, 1056, and 1058 are executed as
26 previously described. Otherwise, the program flow returns to step 1017.

27 If the communication microcontroller receives a begin vend session response from the
28 remote server, program flow proceeds to step 1015. This step tests whether the vend session
29 request was accepted by the remote server. If the request was not accepted then step 1013 is
30 executed. At step 1013 a message is sent to the application microcontroller 230.
31 Communication microcontroller 220 sets write/read line 236 high indicating that data is
32 flowing from microcontroller 220 to microcontroller 230 over bus 238. Line 234 sends clock
33 signals from microcontroller 220 to clock data into microcontroller 230. The content of this
34 message is the reason why the vend session request was denied. Typical reasons would be an

1 invalid password or an invalid login. Microcontroller 230 passes the content of this message
2 to microcontroller 160 using line 252, resister 168, buffer 164, and line 162 as previously
3 described. Microcontroller 160 then causes the reason to be displayed on a display device
4 connected to connector 266 using bus 278 and 276 as previously described.

5 Returning to step 1015, if the vend session request was accepted program flow
6 proceeds to step 1011 where a breeding message is displayed on the display device connected
7 to connector 266 following the same process previously described for displaying messages on
8 this device. A typical greeting message would be "Hello" then the account name, and
9 "Welcome back." The greeting message and the amount of funds available for the account
10 are returned as part of the vend session accepted message sent by the remote server.

11 Program flow proceeds to step 1020 where application microcontroller 230 requests a
12 vend session from the vending machine using the MDB that connects to connector 264.
13 Microcontroller 230 sets line 166 high so that optical isolator 170 is enabled. Microcontroller
14 230 sends logic voltage signals over line 252 to optical isolator 170 which converts the
15 voltage levels to current and then line 174 carries the current signals to connector 264. Part
16 of this message indicates the amount of funds available to the user of the vending machine.

17 Program flow proceeds to step 1024 where application microcontroller 230 waits for
18 the vending machine to indicate a vend purchase. The vending machine sends signals over
19 the MDB onto line 172 and to optical isolator 260. Isolator 260 converts the current pulses
20 into voltage levels that are carried by line 250 to microcontroller 230. If the microcontroller
21 does not receive a vend purchase signal from the vending machine, program flow moves to
22 step 1050 is indicated by the no branch. Step 1050 performs the cancel button test as
23 previously described. If the cancel button was not pushed then the program flow moves to
24 step 1060. At step 1060 communication microcontroller 230 checks to see if a timeout notice
25 was sent by the remote server. If a timeout message has not been received program flow
26 returns to step 1024. If a timeout message is received then steps 1054, 1056, and 1058 are
27 executed as previously described. Returning to step 1050, if the cancel button is pushed then
28 steps 1054, 1056, and 1058 are performed.

29 If the application microcontroller receives a vend purchase signal from the vending
30 machine program flow precedents to step 1026. In this step, application microcontroller 230
31 sends a vend purchase request to the communication microcontroller 220 over bus 238 and
32 lines 232 and 236 as previously described. In step 1028 communication microcontroller 220
33 sends the purchase request to the remote server using lines to 212, 214, 216, and 218 through
34 connector 210 as previously described.

1 Communication microcontroller 220 then waits for a vend purchase request response
2 as depicted in step 1030. If a response is not receive the program flow proceeds to step 1050
3 and then either returns to step 1030 or to start as previously described.

4 If a response is received program flow proceeds to step 1032. This step tests if the
5 vend request was approved by the server. If the request was not approved then step 1034 is
6 performed and communication microcontroller 220 sends a message to application
7 microcontroller 230 as previously described. Application microcontroller 230 sends a
8 message to display microcontroller 160 which displays the reason for the purchase being
9 denied. A typical reason is insufficient funds. In step 1036 application microcontroller 230
10 sends a message to the vending machine over the MDB that the vend has failed. Program
11 flow then returns to step 1024 to allow the user to attempt a different purchase.

12 Returning to step 1032 and assuming the vend purchase request was approved by the
13 server, program flow proceeds to step 1038 where application microcontroller 230 authorizes
14 the vending machine to vend product. At step 1040, the application microcontroller waits for
15 a successful vend message over the MDB from the vending machine. If a successful vend
16 message is received then program flow moves to step 1024 in preparation for another vend.
17 If a successful vend message is not received, program flow proceeds to step 1044 where
18 application microcontroller 230 sends a credit request to communication microcontroller 220
19 as previously described. In step 1046 communication microcontroller 220 sends the credit
20 request to the remote server. In step 1048 the communication microcontroller waits for a
21 received credit request response from the remote server. If the credit request response is not
22 received program flow moves to step 1050. At step 1050 the cancel button test is performed
23 as previously described if the cancel button is not pushed then the program flow returns to
24 step 1048 as indicated by step 1052. If the cancel button is pushed steps 1054, 1056, and
25 1058 are performed as previously described. If the credit request response is received from
26 the remote server, program flow moves back to step 1024 in preparation for another vend
27 request from the user.

28 Many vending machines do not have the ability to record when a product is dispensed
29 and when money is deposited in the machine. Consequently, in an alternative embodiment, a
30 different Internet node application can interface to these type of machines to monitor motor
31 activity and payment information. Motor activity indicates when a product is dispensed from
32 the machine and payment information, such as a bill changer or coin changer, indicates how
33 much money was deposited into the machine. One method of interfacing to machines of this
34 type would be to use an interface board that is unique to each machine. This would allow an

1 Internet node targeted toward vending machines to interface with many different types of
2 machines by combining the Internet node with the machine specific interface board. In this
3 way, vend and payment information could be collected from vending machines that are not
4 designed to collect this information.

5 Fig. 7 is a block diagram of an Internet gate of one embodiment of the present
6 invention. The Internet gate is used to connect one or more Internet nodes to the Internet.
7 The primary functions of the Internet gate are to provide a physical interface to the Internet
8 for one or more Internet nodes and to process part of the Internet communication protocol
9 messages so that multiple Internet nodes may be connected to one Internet gate. Internet
10 nodes utilizing either a TTL level RS-232 interface or an RS-485 interface can be connected
11 to this Internet gate. Those skilled in the art will realize that a TTL level RS-232 interface
12 uses TTL voltage levels and RS-232 timing and handshake protocols. Alternatively, a digital
13 data radio could be connected to the Internet gate using the TTL RS-232 interface. The
14 digital data radio will then be used to communicate with one or more Internet nodes. The
15 Internet gate employs two methods of forming a physical connection to the Internet. One
16 method uses a modem and the other method uses an RS-232 port. An alternative
17 embodiment could use a local area network such as a 10/100 base T network for the Internet
18 connection.

19 The system comprises connector 310 that connects to level converter 319. Level
20 converter 319 operates to convert RS-232 voltage levels present at connector 310 into TTL
21 voltage levels for use by the remainder of the system, and vice versa. Level converter 319 is
22 connected to connector 310 via lines 316 and 318. Level converter 319 is connected to
23 microcontroller 320 through lines Receive Data (RXD) 312, and Transmit Data (TXD) 314.
24 As indicated by the arrows, line 316 carries RS-232 level data from connector 310 to level
25 converter 319 and line 312 carries TTL level data from level converter 319 to microcontroller
26 320. Likewise, line 314 carries TTL level data from microcontroller 320 to level converter
27 319 and line 318 carries RS-232 level data from level converter 319 to connector 310. These
28 lines form a serial interface between microcontroller 320 and a serial device connected to
29 connector 310. This interface is used to configure the Internet gate. Typical configuration
30 information may include the number of Internet nodes to search for, if the Internet gate is
31 programmed to search for Internet nodes, and the telephone number a system modem should
32 use for dialing.

33 Microcontroller 320 uses data bus 338 and control bus 336 to communicate with a
34 Universal Asynchronous Receiver Transmitter (UART) 330. Microcontroller 320 contains

1 RAM/ROM program networking memory 321 that holds the software code and working
2 memory for this and other tasks. Crystal 322 provides the master clock for microcontroller
3 320 and UART 330. Microcontroller 320 contains an internal oscillator and crystal 322 sets
4 the reference frequency of the oscillator. Line 324 carries the reference frequency to UART
5 330. Microcontroller 320 is commercially available from Atmel Corporation under the
6 designation AT90S8515-8AC. UART 330 is commercially available from Texas Instruments
7 under the designation TL16C552. An alternative embodiment would combine the functions
8 of both the microcontroller and the UART into one microcontroller.

9 Bus 338 is an 8-bit data bus used for communication between microcontroller 320 and
10 UART 330. Bus 336 is a control bus used for communication between microcontroller 320
11 and UART 330. These two buses form a standard interface between microcontroller 320 and
12 UART 330 that is well understood and has been of the prior art for many years.

13 Power supply 340 creates a regulated DC voltage from a higher unregulated voltage
14 provided by an external power supply (not shown), the regulated voltage is sufficient to
15 operate the various electronic components depicted in Fig. 7. Lines 342 distribute the
16 regulated voltage to the various components. Reset device 328 monitors the voltage on lines
17 342 and generates a reset signal if the voltage is not within the correct tolerance. The reset
18 device is commercially available from Motorola under the designation MC34064D-5.

19 UART 330 can communicate with four serial devices using two built-in serial ports.
20 Because UART 330 has only two serial ports, only two serial devices may be used at one
21 time. Connected to UART 330 are a modem 361, a RS-232 serial port comprised of level
22 translator 362 and connector 372, a TTL level port comprised of connector 384, and a RS-
23 485 port comprised of RS-485 transceiver 386 and connector 388. Bus 350 connects one
24 serial channel of UART 330 to modem 361 and level converter 362. Level converter 362 can
25 only be used if modem 361 is not present and thus, this channel of the UART can use the RS-
26 232 interface only if the modem is not installed. As will be appreciated by one of ordinary
27 skill, the serial port or the modem port could also be selected by providing a software
28 selection option. Bus 350 contains the standard serial lines RXD, TXD, CTS, RTS, DTR,
29 DSR, and RI. These lines are standard for RS-232 interfaces and are well-known in the prior
30 art. If a modem is not installed then microcontroller 320 sets line 352 high so that level
31 translator 362 is turned on. If line 352 is low, then level translator 362 is disabled and has no
32 affect on bus 350. TTL level RS-232 signals are carried by bus 350 between UART 330 and
33 level translator 362. Bus 370 carries the RS-232 level signals between translator 362 and
34 connector 372. Translator 362 performs the function of translating between the TTL level

1 signals on bus 350 and the RS-232 level signals on bus 370. Level translator 362 is
2 commercially available from Linear Technology under the designation LTC1337. The RS-
3 232 port can be connected to an external serial device that provides a connection to the
4 Internet.

5 Modem 361 converts the TTL level signals on bus 350 into signals that are
6 compatible with the public telephone network. Bus 368 represents a two wire interface
7 between the modem and a modular connector 364. A two wire interface is a common
8 telephone system connection method and has been of the prior art for many years. Other
9 modem connections well-known to those of ordinary skill are also possible. The modem is
10 commercially available from Connexant, formerly Rockwell, under the designation
11 SFV336ACFH1D5A0P.

12 Connector 384 is a TTL level RS-232 interface that is typically used to connect a
13 wireless data radio to the Internet gate. RS-485 transceiver 386 shares the serial data lines
14 with connector 384. Either a device connected to connector 384 or transceiver 386 can be
15 used with this channel of UART 330, both devices cannot be used at the same time.
16 Typically, if a wireless data radio is connected to connector 384 then transceiver 386 is not
17 used. Conversely, if nothing is connected to connector 384 then transceiver 386 is used.
18 Line 374 carries data from connector 384 or transceiver 386 to UART 330. Line 376 carries
19 serial data from UART 330 to connector 384 and transceiver 386. When microcontroller 320
20 sets line 358 low, the receiver in transceiver 386 is enabled. When microcontroller 320 sets
21 line 360 high the transmitter in transceiver 386 is enabled. Transceiver 386 is commercially
22 available from Linear Technologies under the designation LTC1483CS8. Line 380 carries
23 the positive RS-485 data from transceiver 386 to connector 388. Line 382 carries the
24 negative RS-485 data from transceiver 386 to connector 388. Line 379 is a ground reference
25 supplied to connector 388.

26 The flowchart of Fig. 6 depicts the process followed by the Internet gate to provide
27 Internet access for multiple Internet nodes. The process begins at step 3000. At step 3002,
28 the Internet gate detects whether the modem or the RS-232 serial port interface will be used
29 for Internet connection. To do this, microcontroller 320 sets line 352 low so that RS-232
30 level translator 362 is disabled. Bus 350 is now free to communicate with modem 361.
31 Microcontroller 320 causes an "AT" command to be sent to the modem. This is
32 accomplished by passing data on bus 338 and control signals on bus 336 to UART 330.
33 UART 330 then converts the data into serial data and outputs it on bus 350. Microcontroller
34 320 then waits for a response from modem 361. If the modem sends a response it will travel

1 serially over bus 350 to UART 330. The UART will convert the serial data to parallel data
2 and pass it to microcontroller 320 using buses 338 and 336. The "AT" command is a
3 standard modem command well-known in the prior art. If modem 361 responds, level
4 translator 362 will remain disabled and the Internet gate will communicate through the
5 modem. If modem 361 does not respond then microcontroller 320 assumes that modem 361
6 is not installed. Under this condition, microcontroller 320 sets line 352 high so that RS-232
7 translator 362 is enabled and the RS-232 interface and its associated buses are used for the
8 Internet connection.

9 Once the network connection has been selected, microcontroller 320 establishes a
10 connection to the Internet at step 3004. Program flow proceeds to step 3006 where
11 microcontroller 320 sets variables which are contained in memory 321 and used to keep track
12 of which port is reading TCP/IP packets and which port is echoing TCP/IP packets. The
13 program sets the in-port variable that is contained in RAM/ROM program and working
14 memory 321 to modem or RS-232 interface as determined in step 3002. At step 3008 lines
15 358 and 360 are configured so that connector 384 or the RS-485 interface, comprised of
16 transceiver 386 and connector 388, will be used. If connector 384 will be used, then Line 358
17 is set high to disable the RS-485 receiver and line 360 is set low to disable the RS-485
18 transmitter. Otherwise lines 358 and 360 are configured to use the RS-485 interface.

19 Program flow proceeds to step 3010 where microcontroller 320 waits for data to be
20 received, in this case a valid IP header, from the in-port as determined above. If no data is
21 received program flow moves to step 3034. In step 3034 the in-port and out-port variables
22 contained in RAM/ROM program and working memory 321 are switched. This allows data
23 flow in the opposite direction. The program proceeds to step 3036 where a test is made to
24 determine if the Internet connection has been lost. If the connection is lost and then the
25 connection is re-established as depicted in step 3038. In either case, connection lost and re-
26 established, or connection contact, program flow returns to step 3010 to check again for a
27 valid IP header. The process is ready to relay data from one port to the other port. Assuming
28 that the modem and RS-485 physical connections are used then data can be relayed from the
29 modem to the RS-485 port or vice versa depending on how the in-port and out-port variables
30 are set.

31 Once a valid IP header is received, the program proceeds step 3012 where the type of
32 data following the IP header is checked. If the data type is not a TCP packet then program
33 flow moves to step 3014. In this step the data type is checked to determine if it is an ICMP
34 type. If the data type is not an ICMP type, program flow moves to step 3034 where it

1 operates as previously described. If the data type is an ICMP packet then program flow
2 moves to step 3028 where the data is tested to determine if the ICMP packet is complete. If
3 the ICMP packet is not complete, then program flow moves to step 3034 and the in-port and
4 out-port variables are switched as previously described. If the ICMP packet is complete then
5 program flow moves to step 3030.

6 At step 3030 the program determines if the ICMP packet was an echo packet. This
7 packet type is echoed back to the sender. If the packet was an echo packet then step 3032 is
8 performed. This step sends the received packet, or echoes the packet, back to the current in-
9 port. The Internet gate will only respond to an echo ICMP packet, all other ICMP packets or
10 ignored. Program flow then moves to step 3034. Returning to step 3030, if the ICMP packet
11 was not an echo packet then step 3034 is performed.

12 Returning to step 3012 if the packet type is TCP then program flow moves to step
13 3016. This step tests if the packet is complete and error free. If the packet is not complete or
14 contains errors then program flow moves to step 3034 where the program operates as
15 previously described. If the packet tested at step 3016 is complete and error free then the
16 program proceeds to step 3018.

17 At step 3018 the microcontroller determines if the TCP packet is a synchronization
18 packet. If the packet is a synchronization packet then program flow proceeds to step 3022.
19 At this step, the Internet gate sends a synchronization acknowledge packet back to the current
20 in-port. Step 3034 and the following steps are then performed as previously described. If the
21 TCP packet tested in step 3018 wasn't a synchronization packet then step 3020 is performed.

22 At step 3020, the program determines if the TCP packet is a finish packet. If it is a
23 finish packet, then step 3024 is performed. At step 3024 a finish acknowledge packet is sent
24 back to the current in-port. Then step 3034 and the following steps are performed as
25 previously described. If the TCP packet tested in step 3020 was not a finish packet then step
26 3026 is performed.

27 At step 3026, the TCP packet is echoed to the current out-port. In this way all TCP
28 packets, except for synchronization and finish packets, are passed through the Internet gate
29 from the current in-port to the current out-port.

30 Assume that modem 361 in Fig. 7 is used for connection to the Internet and a data
31 radio is connected to connector 384. Next assume that the TCP/IP packet is received by the
32 modem from the two wire telephone interface of connector 364 and bus 368. Modem 361
33 will convert the data received over the telephone interface into RS-232 data and pass this data
34 over bus 350 to UART 330. UART 330 will convert the RS-232 data into parallel data and

1 pass this data to microcontroller 320 using data bus 338 and control bus 336. Microcontroller
2 320 will examine the data starting at step 3010 in Fig. 6. Test 3010 would determine that the
3 packet is an IP packet and the program would proceed to step 3016. Assuming that the
4 packet is complete and error free program flow moves to test 3018. Assuming that the TCP
5 packet is not a synchronization packet or a finish packet then step 3026 is performed and the
6 packet is sent back over data bus 338 using control bus 336 to UART 330. UART 330 then
7 converts the parallel data to serial data and sends the data over transmit data line 376 to
8 connector 384 and onto the data radio.

9 In this way, the packet was routed from the Internet to a data radio to be transmitted to
10 another data radio connected to an Internet node. The Internet node would then respond to
11 the packet with a response packet and the data radio connected to connector 384 would
12 receive the response packet. The response data packet then follows a reverse route through
13 UART 330 to microcontroller 320 then back to UART 330 and to modem 361. In this way
14 TCP packets, except for synchronization and finish packets, are passed through the Internet
15 gate. This allows multiple Internet nodes to be connected to one Internet gate because the
16 Internet gate handles the synchronization and finish packets and passes other TCP/IP packets
17 to the Internet node or multiple Internet nodes. If the Internet gate passed the synchronization
18 and finish packets to the Internet nodes then all of the Internet nodes connected to the Internet
19 gate would try and respond to the synchronization and finish packets. This could cause
20 collisions when the Internet nodes are trying to send data to the Internet gate. Even if the
21 data from multiple Internet nodes passed through the Internet gate and on to the Internet
22 device expecting a response from the synchronization and/or finish packets, that device
23 would see multiple responses thus causing errors. Because the Internet gate prevents the
24 synchronization, finish, and ICMP echo packets from reaching the Internet nodes, multiple
25 Internet nodes can respond during one TCP/IP session. The Internet nodes examine the
26 packets and only one Internet node will respond as described below.

27 The flowchart of Fig. 5 depicts the process of a serial interface Internet node the
28 present invention sending a web page to a remote device, such as a web browser, and
29 allowing the web page to control the setting of a device connected to the Internet node. With
30 reference to the Internet node of Fig. 2, assume that a data radio is connected to connector
31 110. Note that the data radio is not part of the invention and any device that serves to connect
32 the Internet node to an Internet gate or to some other device that makes a physical connection
33 to the Internet would serve. Either connector 110 or connector 182 could be used for the
34 physical connection to the Internet. At step 2002, a TCP/IP packet would be received by the

1 data radio and the data would travel on line 112 to microcontroller 120. Presumably the
2 packet was forwarded by an Internet gate. The program running in RAM/ROM program and
3 working memory 121 would analyze the packet and determine if the remote device, such as a
4 web browser, is attempting to open the TCP/IP session with the Internet node. If the Internet
5 gate is used for the physical connection to the Internet, then the Internet node does not need
6 to process and respond to synchronization and finish packets as described above. The
7 Internet node will then examine the packet to determine if it should respond, and if so, will
8 send the response to the Internet gate to be forwarded on to the Internet. If some device other
9 than an Internet gate is used for physically connecting to the Internet then the program
10 running in memory 121 will process the synchronization, finish, and ICMP echo packets.
11 However, for the reasons described above, if the Internet node must process these packet
12 tapes because no Internet gate is used, only one Internet node can use the physical connection
13 to the Internet.

14 At step 2004, microcontroller 120 determines if the open TCP/IP session command
15 was successful. If the open TCP/IP session was not successful, program flow returns to step
16 2000 as indicated by step 2006. If the open session was successful, at step 2008
17 microcontroller 120 processes the TCP/IP packet sent by the remote device. The remote
18 device is typically a web browser and the browser will request an HTML web page. If the
19 TCP/IP packet is complete, at step 2010 communications microcontroller 120 will send the
20 packet to application microcontroller 130. This is accomplished by setting line 136 high
21 indicating that data will flow from microcontroller 120 to microcontroller 130. Data flows of
22 over bus 138 and is clocked into microcontroller 130 using line 134. Microcontroller 130
23 analyzes the received packet at step 2012.

24 Microcontroller 130 first determines if the TCP/IP packet is intended for this Internet
25 node at step 2012. If the packet is not intended for this Internet node program flow returns to
26 start at step 2014. If the packet is intended for this Internet node, program flow proceeds to
27 step 2016 where microcontroller 130 analyzes to see if the command is complete and valid.
28 If the command is not complete then program flow moves back to step 2000 as indicated by
29 step 2018. If the command is complete then program flow moves to step 2020.

30 In step 2020, microcontroller 130 determines if an action needs to be performed based
31 on the information contained in the packet. If an action needs to be performed on the
32 connected device program flow proceeds to step 2022 where that action will be performed.
33 Typical actions would involve sending serial data to a device connected to the Internet node.

1 For example, assume that a serial thermostat controller is connected to connector 164.
2 If microcontroller 130 needs to send data to the thermostat controller the following actions
3 would be taken. First, microcontroller 130 would check CTS line 154. This line indicates if
4 the thermostat controller is ready to receive data. The CTS signal from the thermostat
5 controller would be carried over line 172 to level translator 160, which converts the RS-232
6 level signal on line 172 to a TTL level signal carried by line 154. If the thermostat controller
7 is ready to receive data, then TTL level data is sent by microcontroller 130 over line 152 to
8 level translator 160. Translator 160 converts the data to RS-232 levels and it is sent through
9 connector 164 to the thermostat controller. If application microcontroller 130 needs to
10 receive data then RTS line 156 is asserted by the microcontroller indicating that it is ready to
11 receive data. This signal is converted by translator 160 and sent as a RS-232 level signal over
12 line 174 to connector 164 and on to the thermostat controller. RS-232 level data is then sent
13 by the thermostat controller to connector 164 and is carried by line 168 to level translator 160
14 where it is converted to TTL level data and sent over RXD line 150 to microcontroller 130.
15 As will be appreciated by one of ordinary skill, although a thermostat controller was used as a
16 specific example, any serial RS-232 based device would work as described above.

17 Alternatively, if a RS-485 based device is connected to the Internet node instead of
18 the RS-232 based thermostat controller of the previous example, that device would be
19 connected to connector 166. Line 176 would carry the positive RS-485 data between
20 connector 166 and RS-485 transceiver 162. Line 178 carries the negative RS-485 data
21 between connector 166 and RS-485 transceiver 162. The TTL level receive and transmit data
22 are carried between the transceiver 162 and application microcontroller 130 using lines 150
23 and 152 respectively. Line 158 when set low enables the receiver section of transceiver 162
24 and line 160 when set high enables the transmitter section of transceiver 162 with line 179
25 providing a ground reference for connector 166.

26 If interaction with the external device was not required at step 2020, or after the
27 interaction has been completed, program flow proceeds to step 2024. In this step, the current
28 state of the connected device is read from the device. This is accomplished using RXD line
29 150, TXD line 152, CTS line 154, RTS line 156, and control lines 158 and 160 as described
30 above under the control of the program running in microcontroller 130.

31 Program flow proceeds to step 2026 where an HTML web page is constructed based
32 on the status of the connected device. Due to memory constraints, the constructed web page
33 is usually quite simple and typically consists of a small quantity of text and numeric
34 information. The invention is not limited to nongraphical information, but memory

1 constraints currently typically restrict applications to nongraphical information. The HTML
2 web page is then sent to communication microcontroller 120 as indicated in step 2028.
3 Microcontroller 130 sends the HTML web page to communication microcontroller 120 by
4 setting line 136 low, outputting data onto bus 138, and clocking the data into microcontroller
5 120 using line 132. Once communications microcontroller 120 has received the HTML web
6 page, program flow moves to step 2030.

7 In step 2030, communications microcontroller 120 sends the formatted web page
8 received from application microcontroller 130 to TTL level serial connector 110.
9 Alternatively, RS-485 connector 182 could be used. TTL level data is sent on line 114 to
10 connector 110 and to transceiver 180. If the RS-485 port is used, then line 186 is set high,
11 before sending data, so that the RS-485 transmitter of transceiver 180 is enabled. Program
12 flow then moves to step 2032.

13 In step 2032, the close TCP/IP, or finish packet, will be processed if it is received. If
14 the Internet gate is forwarding packets to Internet nodes then the Internet gate will respond to
15 the finish packet. Otherwise, communications microcontroller 120 will process and respond
16 to the finish packet. Program flow then returns to step 2000 as indicated by step 2034.

17 In alternative embodiments, many different devices can be connected to Internet
18 nodes. For example, a home burglar alarm could be connected to a serial Internet node, or an
19 Internet node with a interface unique to a specific type of home burglar alarm could be used
20 to interface with the alarm. Another alternative would incorporate a camera into an Internet
21 node. In this application, a camera could be attached to an Internet node or embedded into
22 the Internet node so that still or video pictures of a remote location from the user could be
23 retrieved. The ability to capture sounds, such as a human voice, or play sounds could also be
24 incorporated into the Internet node.

25 The above description of the invention is intended to be illustrative and not limiting.
26 Other embodiments of this invention will be obvious to those skilled in the art in view of the
27 above disclosure.
28

1 CLAIMS

2 What is claimed is:

3

- 4 1. A system for connecting an appliance to the Internet comprising:
- 5 a first microcontroller programmed to communicate with the Internet;
- 6 a first port for connecting the first microcontroller to the Internet;
- 7 a second microcontroller connected to the first microcontroller and
- 8 programmed to communicate with an appliance to send and retrieve
- 9 information to be used in a web page; and
- 10 a second port for connecting the second microcontroller to the appliance.
- 11
- 12 2. The system of claim 1, the information including status information from the
- 13 appliance.
- 14
- 15 3. The system of claim 1, the information including control information that may be sent
- 16 from the Internet.
- 17
- 18 4. The system of claim 1, the appliance being selected from the group consisting of a
- 19 lamp, a thermostat, a home appliance control system, a burglar alarm, and a vending
- 20 machine.
- 21
- 22 5. A system for connecting a vending machine to the Internet comprising:
- 23 a first microcontroller programmed to communicate with the Internet;
- 24 an Internet port for connecting the first microcontroller to the Internet;
- 25 a second microcontroller connected to the first microcontroller and
- 26 programmed to communicate with a vending machine and send and
- 27 receive purchaser, account, debit and credit information over the
- 28 Internet; and
- 29 a vending machine port for connecting the second microcontroller to the
- 30 vending machine.
- 31
- 32 6. The system of claim 5, the second microcontroller communicating with a vending
- 33 machine MDB.
- 34

- 1 7. The system of claim 6, the vending machine port comprising an optical isolator and a
2 connector.
3
- 4 8. The system of claim 5, further comprising a third microcontroller programmed to
5 control a visual display on the vending machine.
6
- 7 9. A method of connecting appliances to the Internet comprising:
8 providing a first microcontroller for establishing a connection with the
9 Internet;
10 providing a second microcontroller with its own IP address for communicating
11 with an appliance;
12 establishing a connection between the first and second microcontrollers;
13 communicating with the Internet by having the first microcontroller respond to
14 Internet connection related signals, passing all other signals to the
15 second microcontroller, and having the second microcontroller respond
16 to all signals that are addressed to its IP address.
17
- 18 10. The method of claim 9, the step of communicating further comprising having the first
19 microcontroller respond to all synchronization, finish, and ICMP echo packets sent
20 over the Internet to the first microcontroller.
21
- 22 11. The method of claim 9 further comprising providing a third microcontroller with its
23 own IP address for communicating with an appliance, the step of communicating
24 further comprising having the first microcontroller pass all signals not related to
25 Internet connection to the third microcontroller and having the third microcontroller
26 respond to all signals addressed to its IP address.
27
- 28 12. The method of claim 9 further comprising providing a data radio to the first and
29 second microcontrollers for connecting between microcontrollers.
30
31

- 1 13. A system for connecting to the Internet comprising:
2 an Internet gate including a first microcontroller for handling Internet
3 connection protocols;
4 an Internet node including a second microcontroller having its own IP address;
5 the Internet gate including a connection port for connecting the first and
6 second microcontrollers to each other;
7 the first microcontroller programmed to respond to all synchronization, finish,
8 and ICMP echo packets received by it and to pass all other signals
9 received from the Internet to the connection port.
10
- 11 14. The system of claim 13, the connection port comprising a RS-232 serial connection.
12
- 13 15. The system of claim 13, further comprising a first data radio connected to the
14 connection port and a second data radio connected to the second microcontroller, the
15 data radios configured to communicate with each other.
16
- 17 16. The system of claim 13, further comprising a second Internet node including a third
18 microcontroller connected to the connection port, having its own IP address for
19 communicating with a second appliance, and programmed to respond to all Internet
20 communications addressed to it.
21
- 22 17. The system of claim 13, the appliance being selected from the group including a lamp,
23 a thermostat, a home appliance control system, the burglar alarm, and a vending
24 machine.
25

FIG. 1.

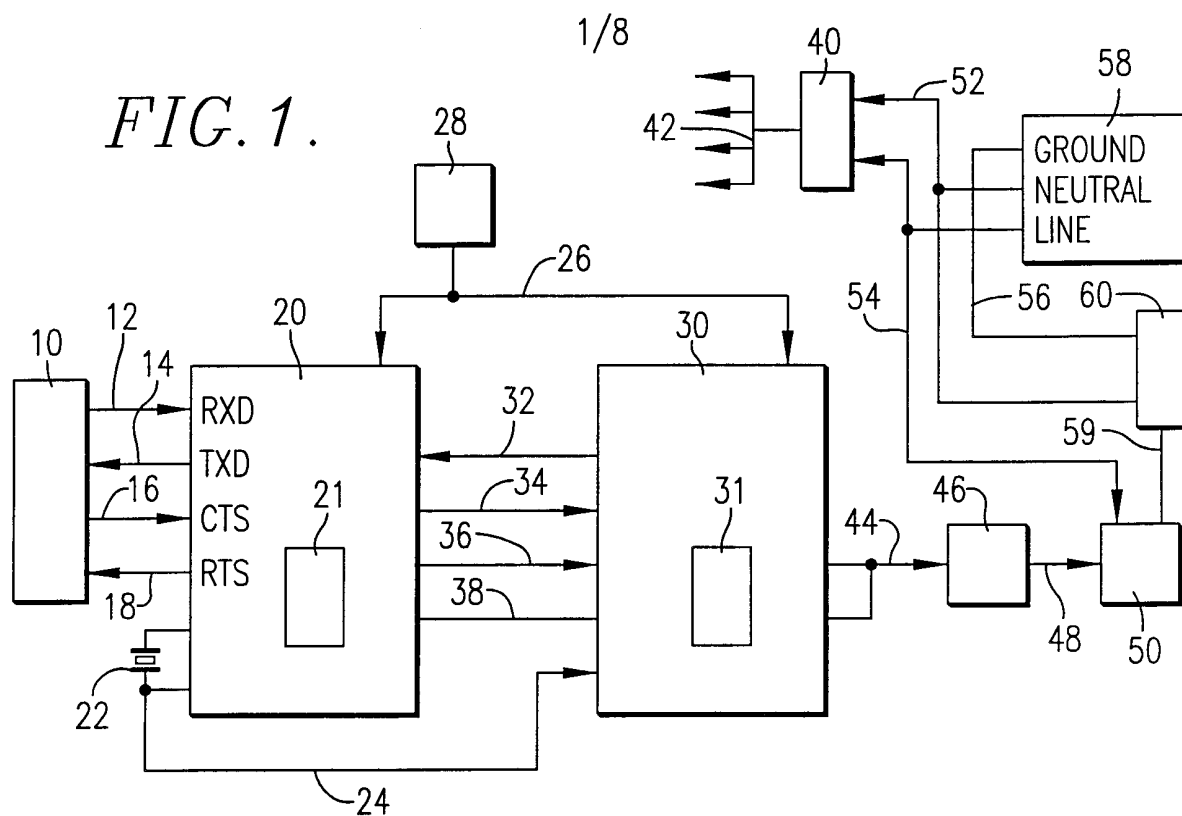


FIG. 2.

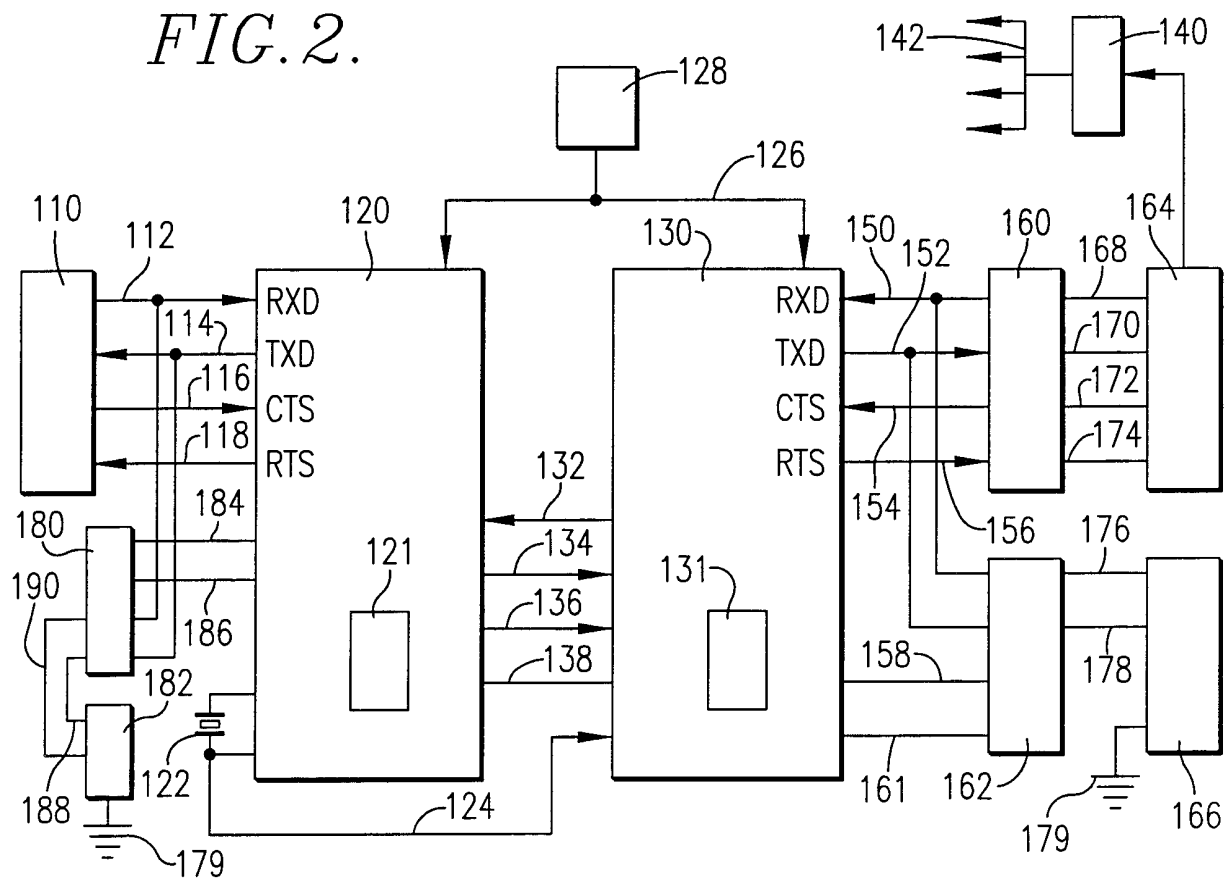


FIG. 3.

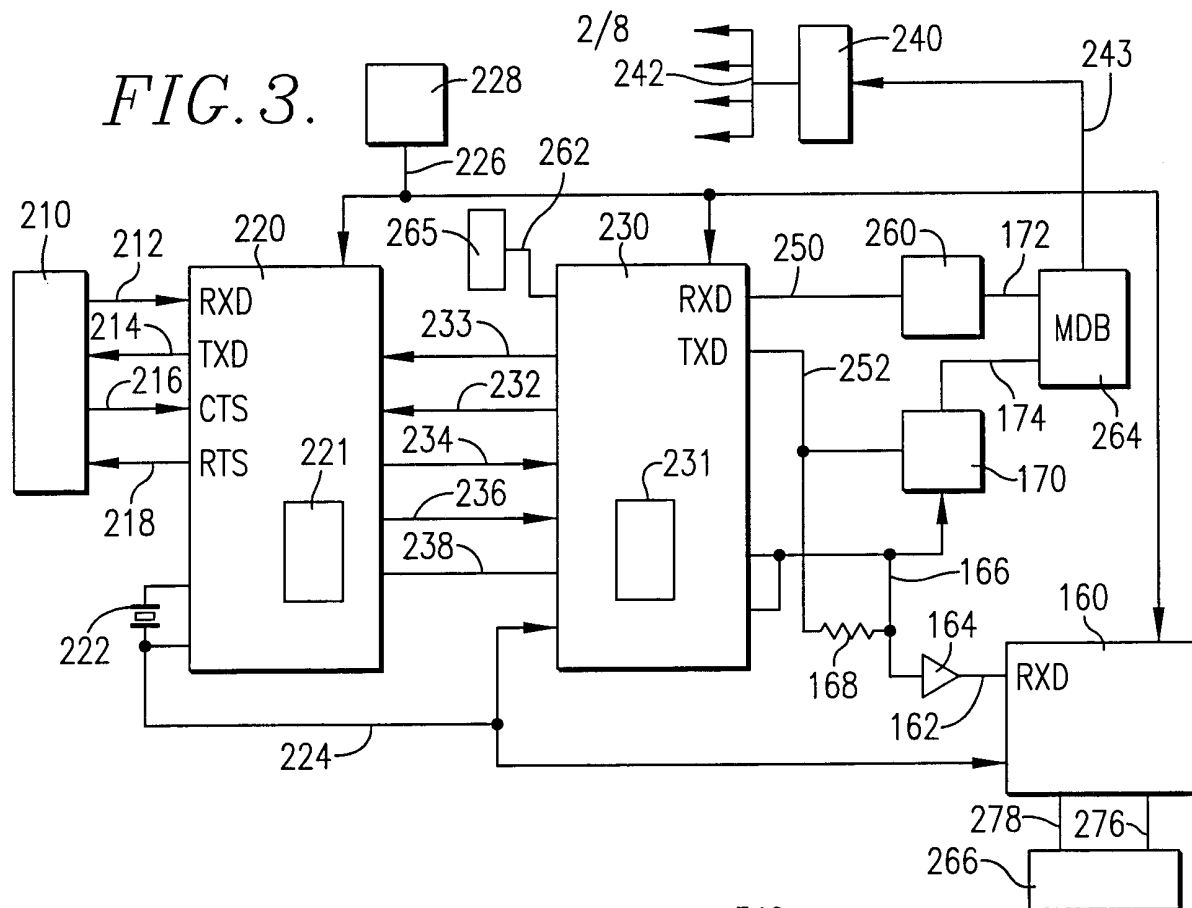
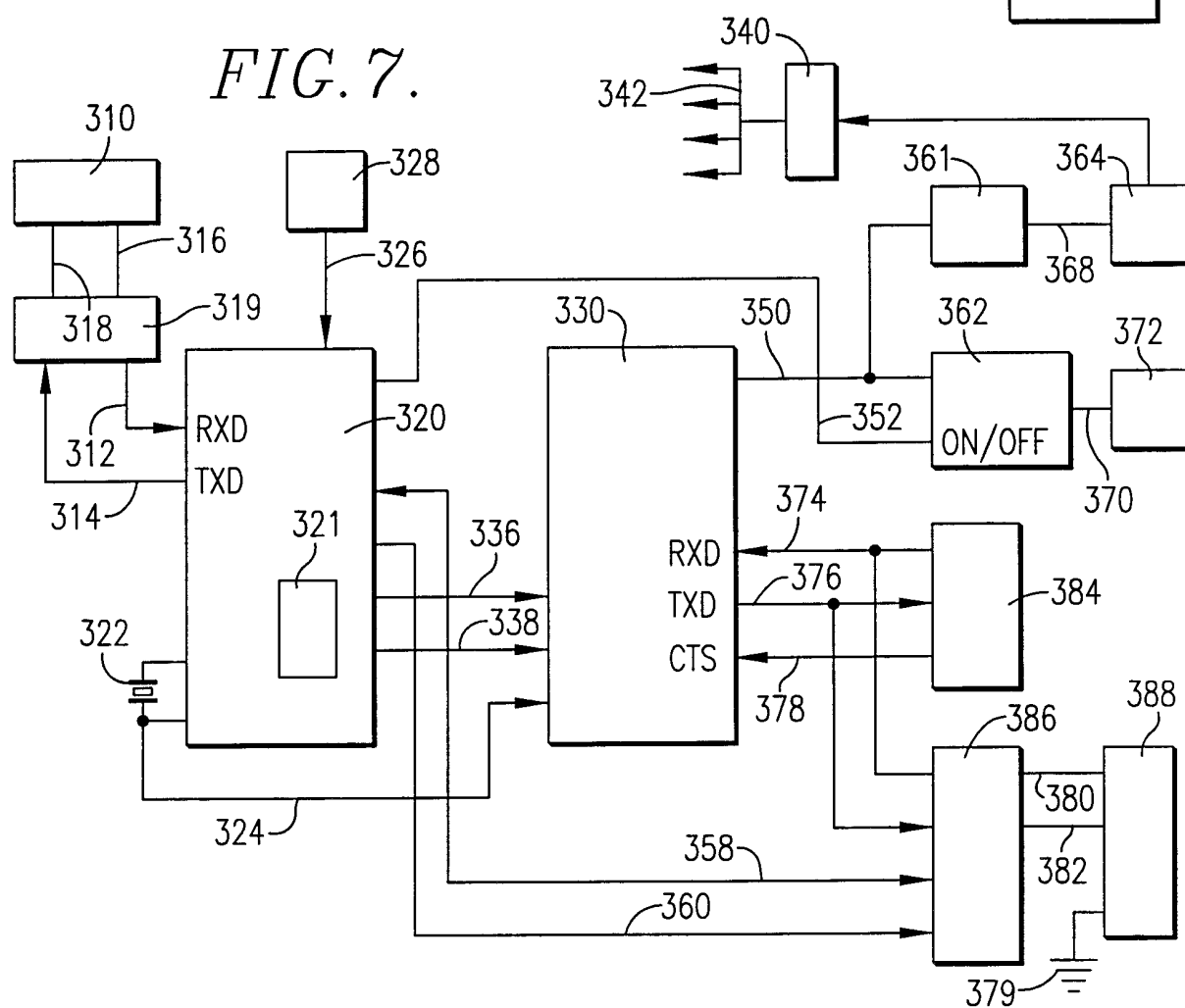
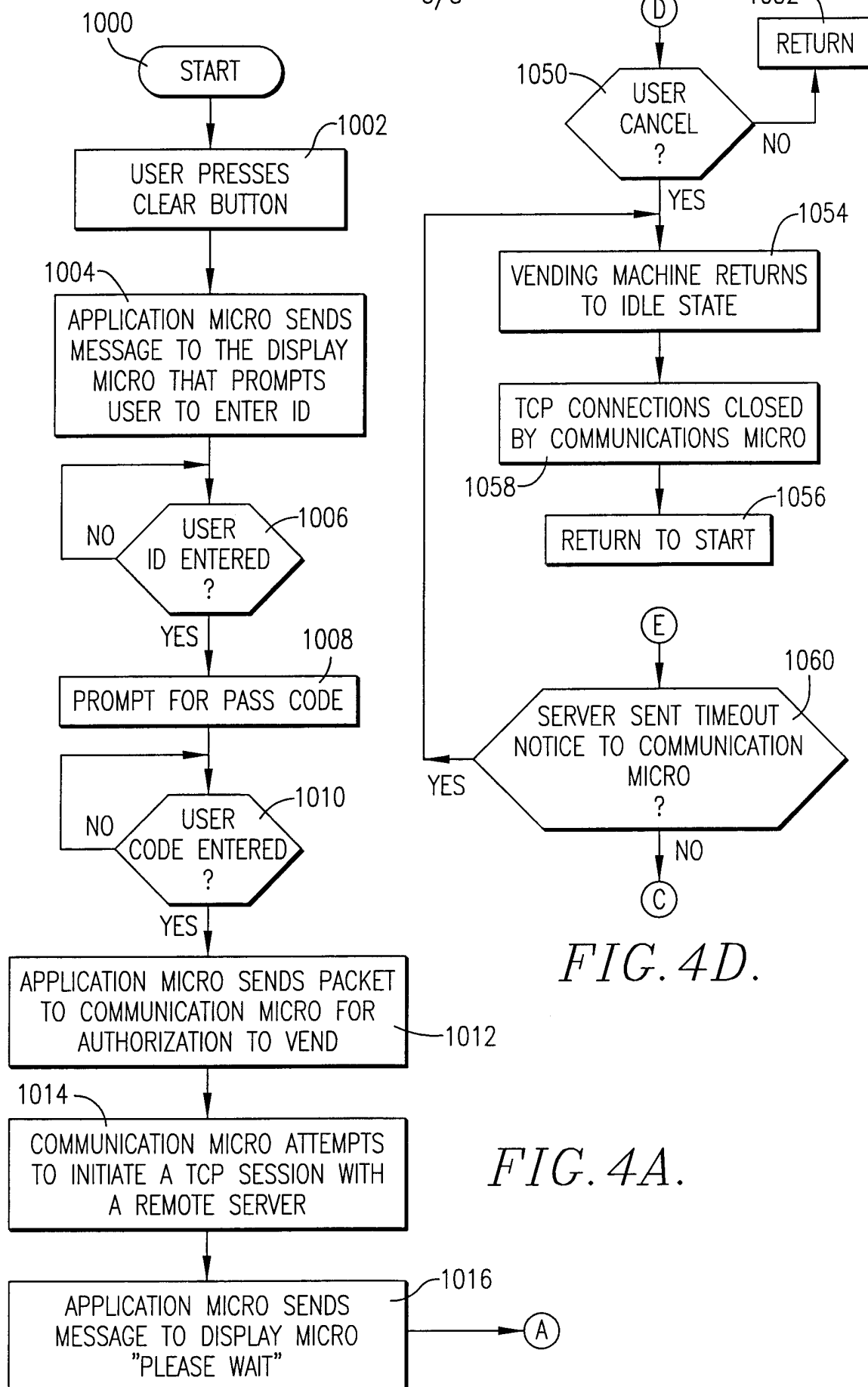
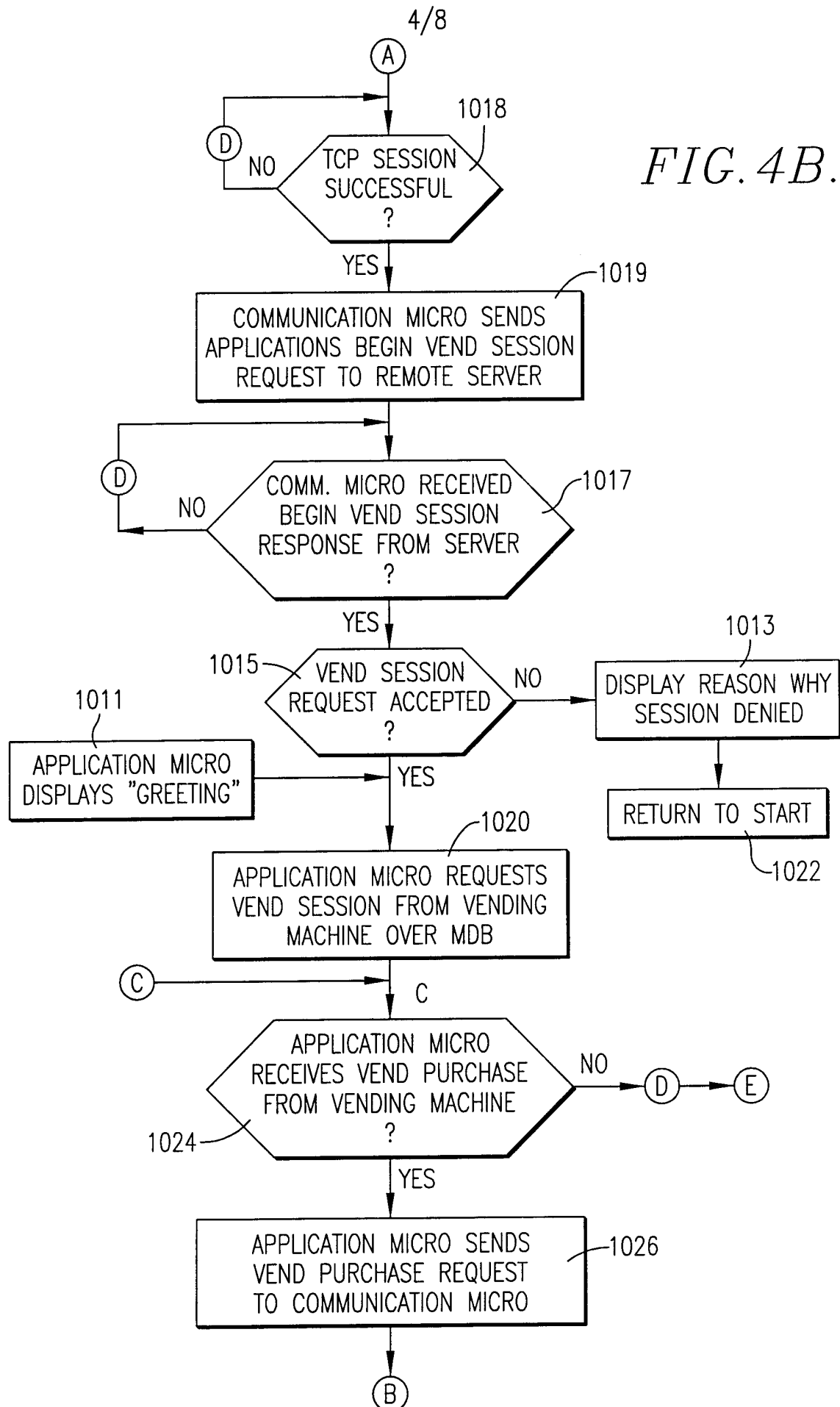


FIG. 7.

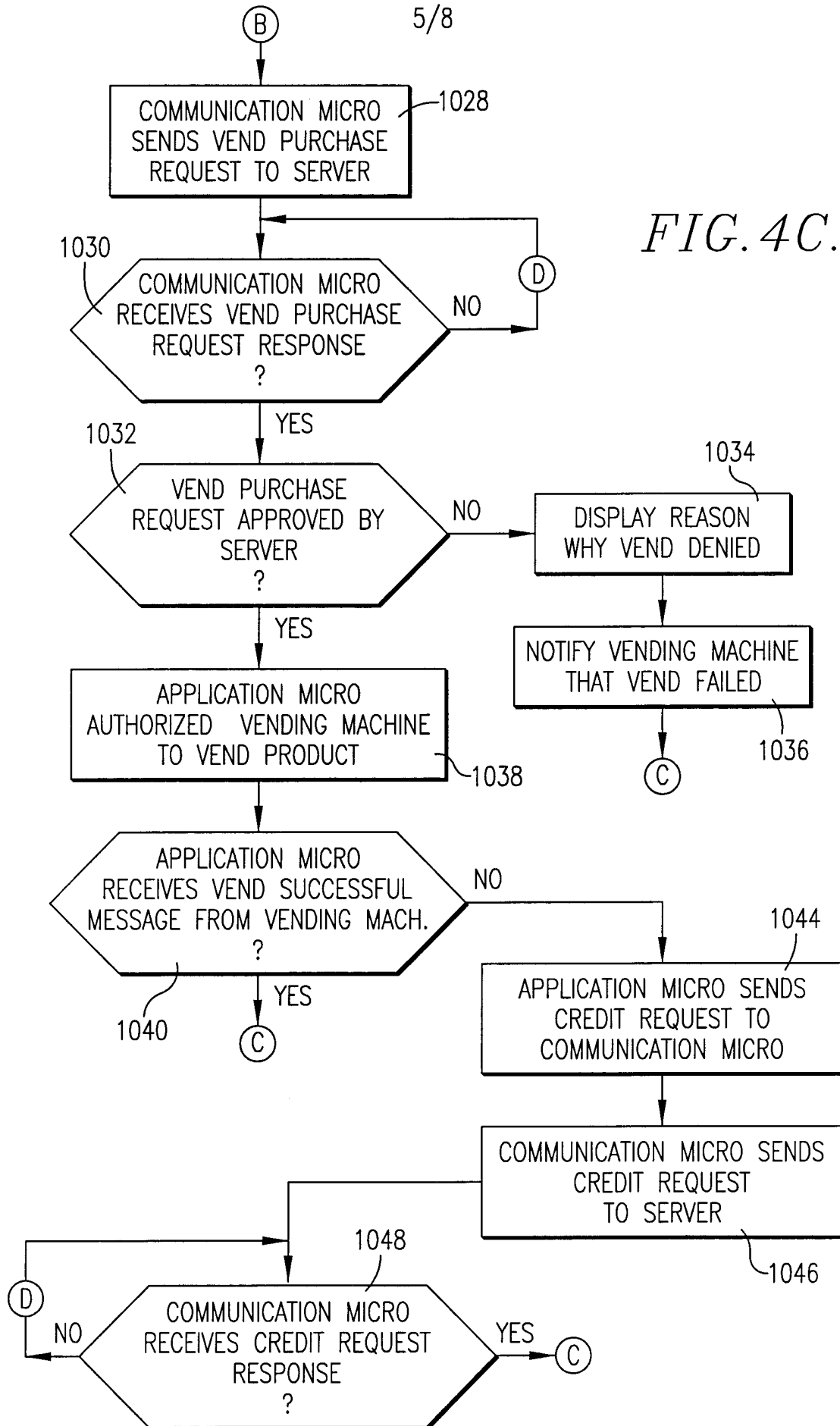


3/8





5/8



6/8

FIG. 5.

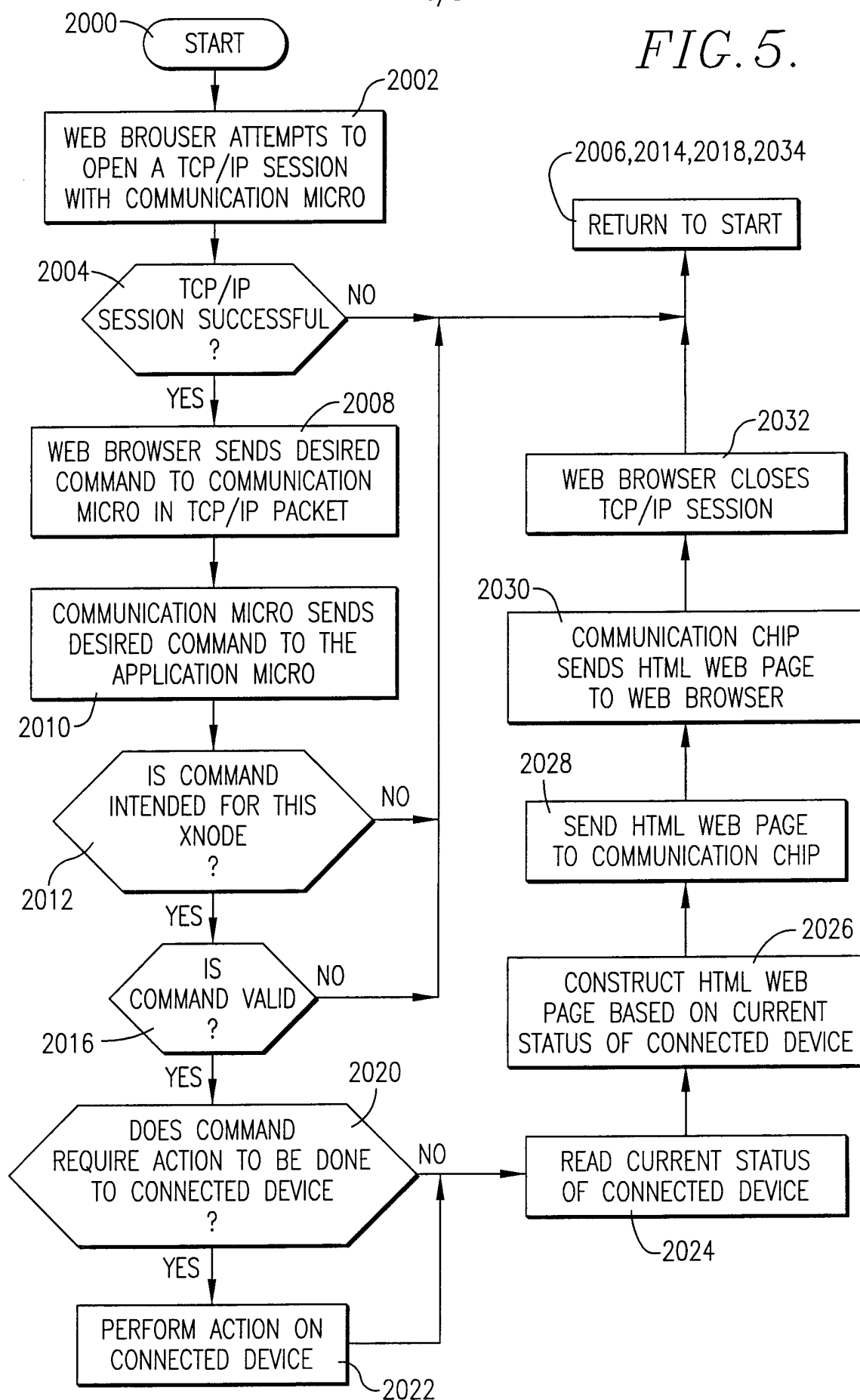
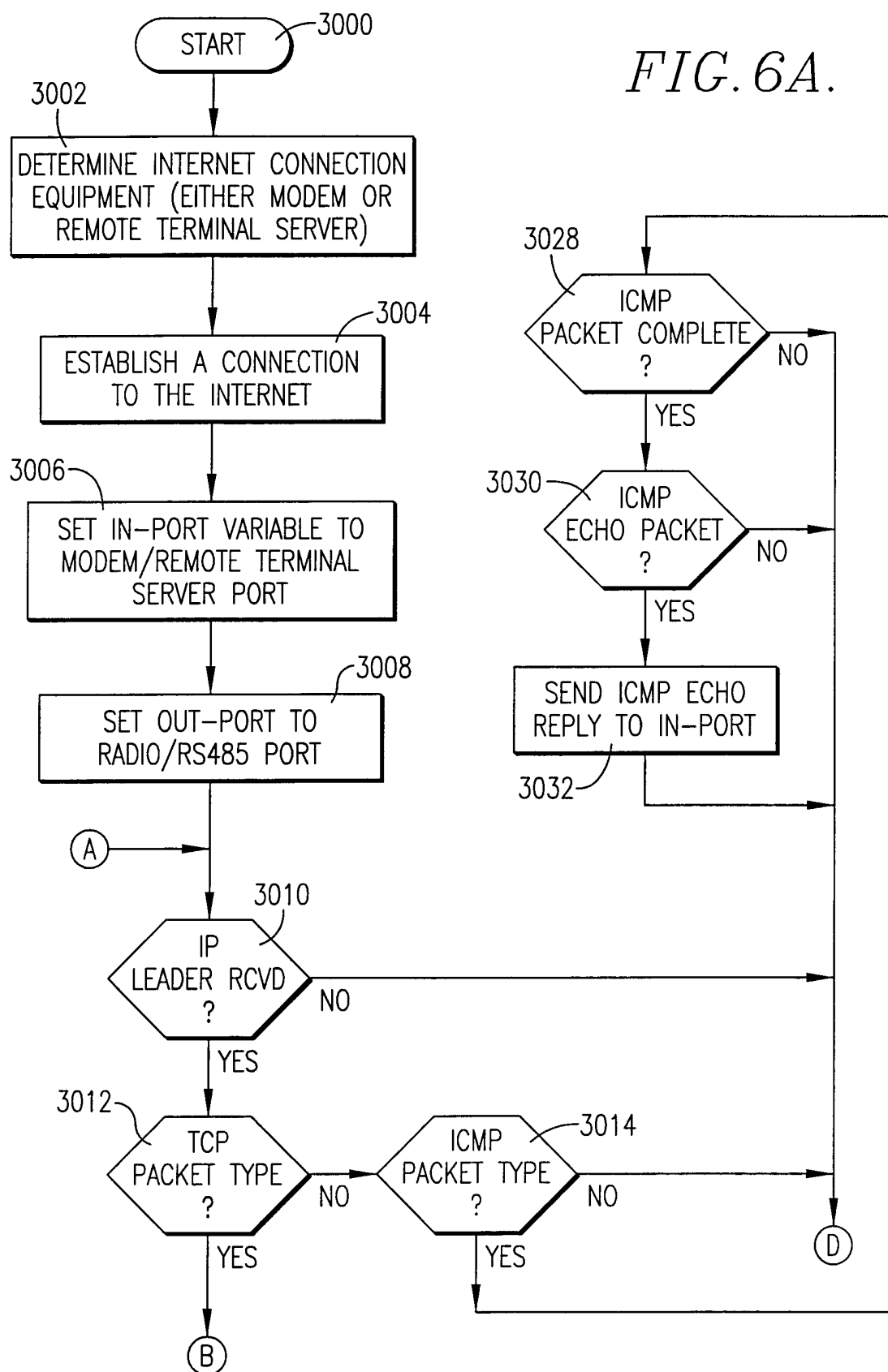


FIG. 6A.



8/8

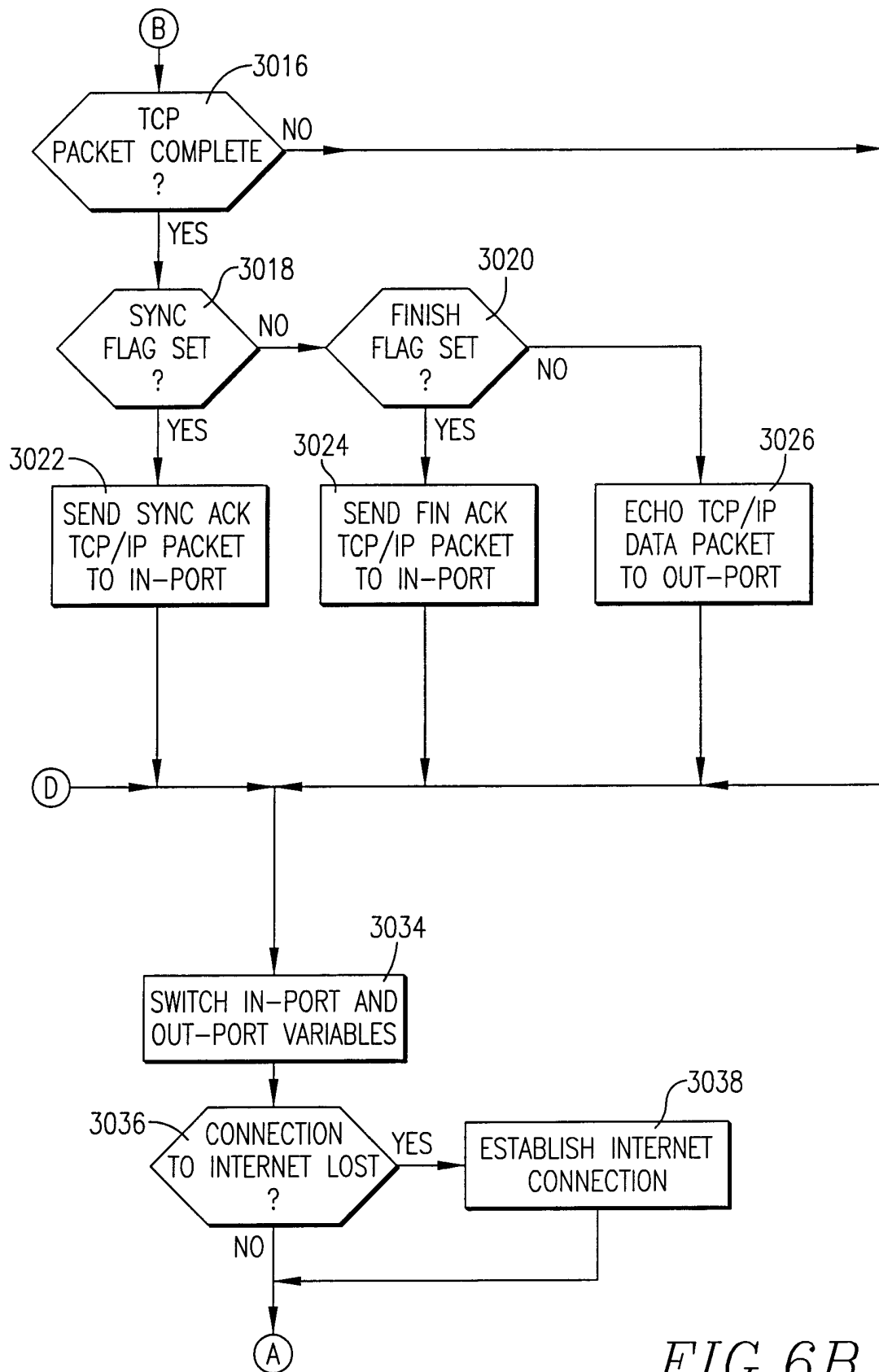


FIG. 6B.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/40323

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 15/173

US CL : 709/224

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)

U.S. : 709/224

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
noneElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
STN/CAS**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,185,737 B1 (NORTHCUTT et al) 06 February 2001, abstract.	1-17
Y, P	US 6,161,133 A (KIKINIS) 12 December 2000, abstract, cols. 2-4.	1-17
Y, P	US 6,124,806 A (CUNNINGHAM et al) 26 September 2000, abstract, cols. 1-3, 7, 9 and 13.	1-17
Y	US 6,021,469 A (TREMBLAY et al) 01 February 2000, abstract.	1-17
Y	US 5,982,445 A (EYER et al) 09 November 1999, abstract.	1-17



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"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)	"&" document member of the same patent family
"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

10 MAY 2001

Date of mailing of the international search report

13 JUN 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

JOHN FOLLANSBEE

Telephone No. (703) 305-8498

James R. Matthews