BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LXI (LXV), Fasc. 1, 2015
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

MANAGEMENT OF AN INTERFACE USING A MICROCONTROLLER FOR RECEIVING CALLER IDENTIFICATION DELIVERY MESSAGES (I)

BY

PETRUȚ DUMA*

“Gheorghe Asachi” Technical University of Iaşi
Faculty of Electronics, Telecommunications and Information Technology,

Received: February 18, 2015
Accepted for publication: March 10, 2015

Abstract. This paper describes the physical specifications for the caller ID, the parameters of signals, sending data on vocal band through a modem, communication principles and protocols, the structure of the messages containing useful information. The hardware structure of the interface for receiving caller ID messages is presented alongside the command program, necessary to the application, which has a range of facilities for displaying the data and storing messages in a nonvolatile memory.

Key words: caller identification delivery messages; continuous phase binary frequency shift keying modulation; single/ multiple data message format.

1. Introduction

In order to implement the caller identification delivery (CID) it is necessary that the called subscriber set is equipped with a specialized interface, in order to receive and display the telephone number of the calling subscriber. The Bellcore laboratories have prepared the specifications and described the characteristics and functionalities for the equipments/interfaces which are essential for making a caller ID system.

* *e-mail*: pduma@etti.tuiasi.ro

The caller number delivery (CND) service sends the information with the number of the calling subscriber between the first burst and the second burst of the ring. The caller name delivery (CNAM) service sends the number and the name of the person who is calling. Another service improves the CND with the waiting call service, so that the subscriber who is engaged in a call can see who is ringing him.

Currently, there are implemented specialized integrated circuits that receive the information for caller ID using the modulation frequency shift keying (FSK), based on the Bell 202 or ITU-TU-23 standards.

The basic structure of an interface for receiving caller ID information that equips the called subscriber telephone set is illustrated in Fig. 1. It shows an interface for the telephone line (ITL) and an interface for receiving caller ID messages (IRCIDM), which are managed by a control unit (CU), implemented with a development system equipped with an ATMEL microcontroller (DS-μc) or with an application system equipped with a PIC microcontroller (AS-μc). The telephone line (TL) is connected to the interface through a C_TL connector, while for making telephone calls, the telephone (T) set is linked using a C_T connector.

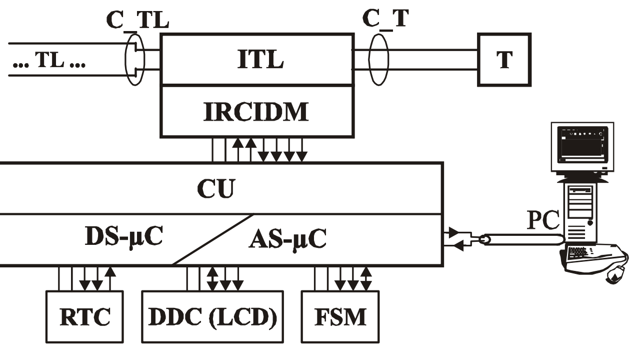

Fig. 1.

The interface for the telephone line receives the call signal, monitors the subscriber line, but also performs other functions which are not the object of this application. The interface for receiving caller ID messages contains an FSK decoder, which provides the useful information through a serial interface.

All the caller ID messages are saved in the FLASH serial memory (FSM). The data displaying console (DDC) shows on an LCD, for each incoming call, the date and time, the calling number, the name of the calling subscriber and number of calls. When no messages or other information are to be shown on the LCD, the time from a real time clock (RTC) is shown.

This paper presents the functional characteristics for caller ID, according to the Bellcore references. Another work will elaborate on the hardware structure and the command program for managing the interface that receives caller ID messages.

2. The Physical Level Specifications for Caller ID

The Bellcore references for ensuring the caller ID service specify the following parameters of the physical level from the central office (CO): link type with two wires that work half-duplex; the used modulation is the continuous phase binary frequency shift keying type; the “1” logic data bit is sent with a sinusoidal signal with the frequency of 1,200 Hz (±1%); the “0” logic data bit is sent with a sinusoidal signal with the frequency of 2,400 Hz (±1%); sending data for identifying the calling subscriber is done with a binary asynchronous serial interface, which runs at a baud rate of 1,200 bits/s (±1%); the signal level is –13.5 dBm ±1.5 dBm at the point of application to the loop facility into a resistive load of 900 Ω; the impedance of the power source is 900 Ω in series with 2.16 μF to meet return loss requirements specifications.

The interface for the caller ID service from the central office must send the necessary data and interacts adequately with the interface of the called subscriber. In this case, the following parameters at the physical interface level for customer premises equipment (CPE) are to be met: link type, modulation type, the coding of data bits, the serial interface and baud rate must be similar to the values sent from the central office that were presented previously; the level of the received signal with the frequency of 1,200 Hz for the “1” logic data bit is between –32 dBm and –12.5 dBm; the level of the received signal with the frequency of 2,400 Hz for the “0” logic data bit is between –36 dBm and –12.5 dBm; the signal to distortion ratio must be greater than 25 dB.

The succession of signals sent by the caller ID between the first call burst and the second burst is indicated in Fig. 2; the notations used have the following meanings: FRB – first ring burst (A), with the duration of 2 seconds or less; P1 – pause (B) with the duration of at least 0.5 seconds after the ending of the first ring burst and before sending the FSK signal; CSS – channel seizure signal (C); MS – mark signal (D); DP – data packet (E); P2 – pause (F) with the duration of at least 0.2 seconds after sending the data packet and before the sending of the second call burst; SRB - second ring burst (G); P3 – pause (H) between the following two ring bursts; NRB – next ring burst (I).

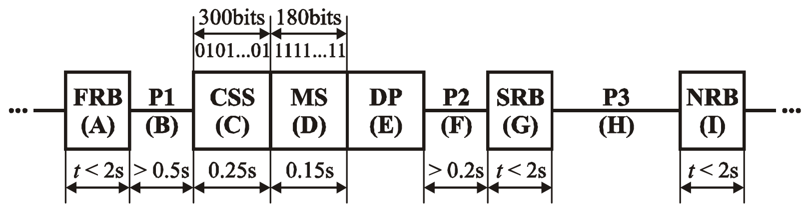

Fig. 2.

After at least 0.5 seconds from sending the first call burst, the sequence of sending the signal for caller ID starts; this begins with the channel seizure signal, which consists of a continuous string of 300 bits, alternating the "0" logic bit with the "1" logic bit. The string always starts with a "0" logic bit and ends with a "1" logic bit and has a duration of 0.25 seconds. Immediately after CSS, a mark signal is sent, consisting of a string of 180 bits with the value of "1" logic and having duration of 0.15 s. The purpose of sending CSS and MS is to prepare the reception of data in CPE.

The data packet with the caller ID message is sent immediately after CSS and MS at a baud rate of 1,200 bits/s. The bytes in the message are sent one after the other in the order of which they are met in the message and they have different significance; some bytes indicate: the type of the message, the length of the message, the type of parameter, the length of the parameter, while the rest of the bytes from the message or from the parameters are ASCII codes of the displayable characters with the usable information of time, date, number or name of the caller.

Every byte from the data packet is sent starting with the less significant bit (LSB), finishing with the most significant bit (MSB). Any sent byte is assembled in a format consisting of a START bit, which is always defined with a "0" logic, the 8 data bits and a STOP bit, defined always with a "1" logic (Fig. 3). The data from the packet is sent continuously without pauses between the bytes. If the central office have no anymore data to send or it waits for the information to become available during a data packet, then 10 bits with the value of "1" logic (at most) can be inserted between the words.

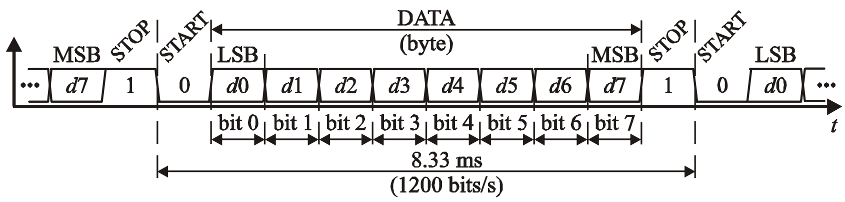

Fig. 3.

3. The Structure of Data Messages for Caller ID

The data interface from the stored program controlled switching system (SPCS) supports single or multiple data message formats.

The single data message format (SDMF) consists of a data string which are sent to the CPE as in Fig. 4; the used notations have the following meaning: SMT – SDMF message type; SML – SDMF message length; MB – message bytes (MB_1, MB_2, ….. MB_N); CS – checksum message SDMF.

The type of message is a numeric value of a byte with the header of the SDMF data packet. The significances of the primary bytes with different SDMF message types are shown in the first lines of Table 1.

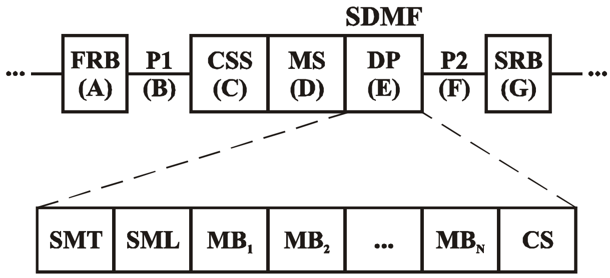

Fig. 4.

Table 1

Byte	Type Message Significance	Format
04H	SDMF packet header	SDMF
06H	Message waiting indicator	SDMF
0BH	Reserved (for message desk information)	SDMF
80H	MDMF packet header	MDMF
81H	MDMF test sequence packet header	MDMF
82H	Message waiting notification	MDMF

The length of the SDMF message is equal to the number of bytes from the message and has a one byte numerical value which does not include the type of message, the length and the checksum.

The SDMF data message bytes contain the information which the central office sends to the end-user. In this case, the useful information contains the bytes for the date (month and day) and for the time (hour and minute) when the call is made, and the number of the calling subscriber. All of these bytes from the data message body are the ASCII codes of the displayable characters for date, time and calling number.

It should be pointed out that CO does not send the number of the calling subscriber if it is blocked by the respective subscriber. In this case, the CPE receives the message with the date, time and the “P” character (50H - ASCII code) in order to indicate the blocking of this number at the request of the subscriber (the private number belongs to the subscriber which does not want to make it a public number). Another situation arises when the calling subscriber is outside the usage area of the phone network. For this situation the CPE receives the message with the time, date and the “O” character (4FH - ASCII code) in order to indicate the out of area situation.

The last byte of the data packet is the checksum, which has the role of detecting the errors in the received message.

An example of a SDMF message is:

Date:	**January 24 (01.24)**
Time:	**10:27**
Number:	**0232.345678**

The structure of the CID message in the single format for this example is shown in more detail in Table 2.

Table 2

Byte order	SDMF message byte type	Character	Byte	Binary value	Decimal value
1	SMT – SDMF Message Type	–	04H**	00000100B	04
2	SML – SDMF Message Length	–	12H**	00010010B	18
3	MB_1 – First figure of the month	0	30H*	00110000B	48
4	MB_2 – Second figure of the month	1	31H*	00110001B	49
5	MB_3 – First figure of the day	2	32H*	00110010B	50
6	MB_4 – Second figure of the day	4	34H*	00110100B	52
7	MB_5 – First figure of the hour	1	31H*	00110001B	49
8	MB_6 – Second figure of the hour	0	30H*	00110000B	48
9	MB_7 – First figure of the minute	2	32H*	00110010B	50
10	MB_8 – Second figure of the minute	7	37H*	00110111B	55
11	MB_9 – First figure of the number	0	30H*	00110000B	48
12	MB_{10} – Second figure of the number	2	32H*	00110010B	50
13	MB_{11} – Third figure of the number	3	33H*	00110011B	51
14	MB_{12} – Forth figure of the number	2	32H*	00110010B	50
15	MB_{13} – Fifth figure of the number	3	33H*	00110011B	51
16	MB_{14} – Sixth figure of the number	4	34H*	00110100B	52
17	MB_{15} – Seventh figure of the number	5	35H*	00110101B	53
18	MB_{16} – Eighth figure of the number	6	36H*	00110110B	54
19	MB_{17} – Ninth figure of the number	7	37H*	00110111B	55
20	MB_{18} – Tenth figure of the number	8	38H*	00111000B	56
21	CS – CheckSum SDMF message	–	51H**	01010001B	81

* – ASCII code ** – hexadecimal value

The multiple data message format (MDMF) consists of a data string which is sent to the interface of the called subscriber, as shown in Fig. 5; the used notations have the following meaning: MMT – MDMF message type; MML – MDMF message length; PM1, PM2, PM3 – the message parameters; MPM – more parameter message; CS – checksum message MDMF.

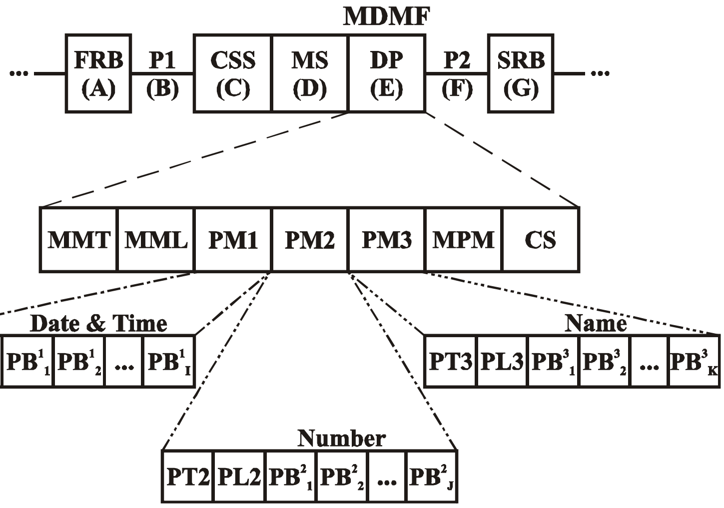

Fig. 5.

Every message parameter is consists of 3 fields: PT – parameter type (PT1, PT2, PT3, …..) has a numerical value of one byte and specifies the significance of the parameter bytes field; Table 3 includes the main numerical values with the significance of the parameter types; PL – the length parameter (PL1, PL2, PL3, …..) has a numerical value of one byte and is equal to the number of data bytes that is stored in the parameter bytes field (does not include PT and PL); PB_1^1, PB_2^1, ….. PB^1_I – parameter bytes for PT1; PB_1^2, PB_2^2, ….. PB^1_J – parameter bytes for PT2; PB_1^3, PB_2^3, ….. PB^1_K – parameter bytes for PT_3.

Table 3

Byte	Type parameter significance
01H	Date and time
02H	Calling line identification
03H	Reserved for dialable directory number (DDN)
04H	Reason for absence of DDN
05H	Reserved for reason for redirection
06H	Call qualifier
07H	Name
08H	Reason for absence of name
0BH	Message waiting notification

The type of message is a numerical value of one byte, including the header of the data packet MDMF; the significance of the main bytes for the different types of MDMF messages is shown in the last lines of Table 1.

The length of MDMF message is equal to the number of bytes from the message; it is a numerical value of one byte, which does not include the type of message, the length of the message and the checksum. This counts the number of bytes from the parameter type fields, parameter length and parameter bytes for all of the message parameters. The length of the message is 6+I+J+K for the three message parameters PM1, PM2 and PM3.

The usual parameters used for sending MDMF messages are: date and time, the calling number and the name of the calling subscriber. For each of these parameters are sent: the type of parameter (01H, 02H and 07H), the length of the parameter (I, J and K) and the bytes of the parameter, which are the ASCII codes of the displayable characters for date, time, number and name of the calling subscriber.

The data packet ends with the checksum for detecting the occurrence of errors in the received message. An example of an MDMF message is:

Date:	**January 24 (01.24)**
Time:	**10:54**
Number:	**0232.456789**
Name:	**Daniel Toma**

The structure of the caller ID message in the multiple format for this example is shown in more detail in Table 4.

Table 4

Byte order	MDMF message byte type	Character	Byte	Binary value	Decimal value
1	MMT – MDMF Message Type	–	80H**	00000100B	128
2	MML – MDMF Message Length	–	23H**	00010010B	35
3	PT1 – Parameter Type - Date &	–	01H**	00000001B	01
4	PL1 – Parameter Length	–	08H**	00001000B	08
5	MB^1_1 – First figure of the month	0	30H*	00110000B	48
6	MB^1_2 – Second figure of the month	1	31H*	00110001B	49
7	MB^1_3 – First figure of the day	2	32H*	00110010B	50
8	MB^1_4 – Second figure of the day	4	34H*	00110100B	52
9	MB^1_5 – First figure of the hour	1	31H*	00110001B	49
10	MB^1_6 – Second figure of the hour	0	30H*	00110000B	48
11	MB^1_7 – First figure of the minute	5	35H*	00110010B	53
12	MB^1_8 – Second figure of the minute	4	34H*	00110111B	52
13	PT2 – Parameter Type - Number	–	02H**	00000010B	02

Table 4
(Continuation)

Byte order	MDMF message byte type	Character	Byte	Binary value	Decimal value
14	PL2 – Parameter Length	–	0AH**	00001010B	10
15	MB^2_1 – First figure of the number	0	30H*	00110000B	48
16	MB^2_2 – Second figure of the number	2	32H*	00110010B	50
17	MB^2_3 – Third figure of the number	3	33H*	00110011B	51
18	MB^2_4 – Forth figure of the number	2	32H*	00110010B	50
19	MB^2_5 – Fifth figure of the number	4	34H*	00110100B	52
20	MB^2_6 – Sixth figure of the number	5	35H*	00110101B	53
21	MB^2_7 – Seventh figure of the number	6	36H*	00110110B	54
22	MB^2_8 – Eighth figure of the number	7	37H*	00110111B	55
23	MB^2_9 – Ninth figure of the number	8	38H*	00111000B	56
24	MB^2_{10} – Tenth figure of the number	9	39H*	00111001B	57
25	PT3 – Parameter Type – Name	–	07H**	00000111B	07
26	PL3 – Parameter Length	–	0BH**	00001011B	11
27	MB^3_1 – First letter of the name	D	44H*	01000100B	68
28	MB^3_2 – Second letter of the name	a	61H*	01100001B	97
29	MB^3_3 – Third letter of the name	n	6EH*	01101110B	110
30	MB^3_4 – Forth letter of the name	i	69H*	01101001B	105
31	MB^3_5 – Fifth letter of the name	e	65H*	01100101B	101
32	MB^3_6 – Sixth letter of the name	l	6CH*	01101100B	108
33	MB^3_7 – Seventh letter of the name	–	20H*	00100000B	32
34	MB^3_8 – Eighth letter of the name	T	54H*	01010100B	84
35	MB^3_9 – Ninth letter of the name	o	6FH*	01101111B	111
36	MB^3_{10} – Tenth letter of the name	m	6DH*	01101101B	109
37	MB^3_{11} – Eleventh letter of the name	a	61H*	01100001B	97
38	CS – CheckSum MDMF message	–	99H**	10011001B	153

* – ASCII code, ** – hexadecimal value

The checksum, either in SDMF or MDMF message format, is the two's complement of the modulo 256 sum of all the bytes from all the message's words. CSS and MS are not included in this calculation. All the bytes from the data packet, except for SC, are summed in central office. This unsigned sum is divided by 256. The quotient of the division is discarded while the remainder of the division is the modulo 256 sum of the bytes from the data packet. This one byte value is bit by bit complemented and incremented by one unit. Thus, the checksum is obtained and sent at the end of the data packet.

At receiving the caller ID message in customer premises equipment, it is calculated the modulo 256 sum of all the bytes in the data packet, including the received checksum. If this sum has the numerical value of zero, then the received message is assumed without errors and the useful information is displayed. If the resulting sum is different from zero, then the received message is erroneous and is not taken into consideration. In this situation, there is no possibility of correcting the errors. In the case when a message with errors is received in customer premises equipment, there is no possibility for central office to resend it.

The Bellcore standards recommend the display of the error messages, but the equipments/interfaces producers chose to ignore them, therefore they are not displayed.

4. Conclusions

This paper presents the main physical level characteristics that allow implementing a caller ID system. The caller ID messages are received using a specialized hardware interface, an LCD display and a nonvolatile memory, managed by a microcontroller system.

The received information is used for displaying the date and time of the call, the number of the calling subscriber, his or her name, the number of calls and other information.

The application can be developed and used to trace calls from a period of time, in order to identify the malicious users, to store the information received from messages in a nonvolatile memory in order to redial quickly, to block unwanted calls etc.

In more complex applications that connect FLASH memory of greater capacity, searches can be conducted into a database in order to get and display various information about the calling subscriber. For example, an insurance company may use this service to display relevant information about a client before answering the telephone, a family medicine cabinet can use it in order to display the health status of the patient who is calling or a delivery company can check the status of the calling person's order.

REFERENCES

Aghion C., Ursaru O., *Aplicații practice ale microcontrolerelor*, Edit. PIM, Iaşi, 2009.

Borcoci E., *Sisteme de comutație digitale*. Edit. Vega, Bucureşti, 1994.

Căpățînă O., *Proiectarea cu microcalculatoare integrate*. Ed. Dacia, Cluj-Napoca, 1992.

Ciurea D., *Transmisiuni telefonice*. Edit. MatrixRom, Bucureşti, 2004.

Duma P., *Centrale telefonice electronice*, Edit. MatrixRom, Bucureşti, 1998.

Duma P., *Microcontrolere în telecomunicații*. Edit. TEHNOPRESS, Iaşi, 2007.

Duma P., *Microcontrolerul INTEL 8051. Aplicații*. Edit. TEHNOPRESS, Iaşi, 2004.

Hintz J.K., Tabak D., *Microcontrollers. Arhitecture, Implementation and Programming.* McGraw Hill, New York, 1993.
Ioan L., Niculescu G., *Comutaţie şi rutare în telecomunicaţii*. Edit. MatrixRom, Bucureşti, 2011.
Peatmann B.J., *Design with Microcontrollers*, McGraw Hill, New York, 1998.
Rădulescu T., *Reţele de telecomunicaţii*, Edit. Thalia, Bucureşti, 2002.
Rădulescu T., *Telecomunicaţii*, Edit. Media Publishing, Bucureşti, 1994.
Somnea D., Vlăduţ T., *Programarea în Assembler.* Edit. Tehnică, Bucureşti, 1992.
* * * *ATMEL, Family Microcontroller.* Data Book, 1998.
* * * *ATMEL, AT89S8253 Microcontroller*. Data Sheet, 2005.
* * * *ATMEL, CMOS Flash Memory*. Data Book, 2004.
* * * *BELLCORE, Calling Number Delivery*. TR-NWT-000031, 1992.
* * * *BELLCORE, Customer Premises Equipment Compatibility Considerations for the Voiceband Data Transmission Interface*. SR-TSV-002476, 1992.
* * * *BELLCORE, Calling Name Delivery Generic Requirements.* TR-NWT-001188, 1992.
* * * *BELLCORE, Voiceband Data Transmission Interface Generic Requirements*. GR-30-CORE, 1994.
* * * *BELLCORE, Testing Guidelines for Analog Type 1, 2, and 3 CPE as Described in SR-INS-002726*. SR-3004, 1995.
* * * *HOLTEK, Type I caller ID using the HT9032*. Application Note, 2000.
* * * *HOLTEK, Calling Line Identification Receiver*. HT9032 Data Sheet, 2000.
* * * *MICROCHIP, PICmicro™ Mid-Range MCU Family*. Reference Manual, 1997.
* * * *NIPPON PRECISION CIRCUITS, FSK Decoder and DTMF Receiver*. SM8223A Data Sheet, 2000.

GESTIONAREA INTERFEŢEI PENTRU RECEPŢIONAREA MESAJELOR CID CU MICROCONTROLER

(Rezumat)

Sunt descrise specificaţiile la nivel fizic pentru CID, parametrii semnalelor, transmiterea datelor în banda vocală pe bază de modem, principii, protocoale de comunicare, structura mesajelor cu informaţie utilă. Structura hardware a interfeţei pentru recepţionarea mesajelor CID este prezentată, alături de programul de comandă necesar aplicaţiei care are implementate diverse facilităţi pentru afişarea datelor şi salvarea mesajelor într-o memorie nevolatilă.