MAP27W32
Dynamic Link Library
v 2.00

User's Guide

Rev. 3.00

http://www.map27.com.ar

(blank)

TABLE OF CONTENTS

TABLE OF CONTENTS....3
License Agreement....5
INTRODUCTION....6
For a Better Comprehension:....6
Documentation Coventions....6
Description....7
INSTALLATION NOTES....8
General....8
OCX Format....8
CHAPTER I – LIBRARY USAGE....9
Using the library into your system....9
Function Callback....9
Initializing....10
Cleaning Up....11
CHAPTER II - GENERAL FUNCTIONS....12
M27_GetLibVersion....12
M27_InitialiseMap27....12
M27_CleanupMap27....13
M27_SetRS232Config....14
M27_GetRS232Config....14
M27_SetCallBack....15
M27_GetData....15
M27_SetCodeInfo....16
CHAPTER III – DTE-RU MESSAGES....18
M27_SendStatusMessage....18
M27_SendMessage....19
M27_ClearCall....19
M27_RadioPersonality....20
M27_NumberingInfo....21
M27_Status....21
M27_OperatingConditions....22
M27_NetworkInfo....22
M27_RadioControl....23
M27_SetNewVolume....24

M27_DialledStrings.....*24*
M27_RadioTest.....*25*
M27_MakeRadioSpeechCall.....*26*
CHAPTER IV – EVENTS RU-DTE.....27
EV_ACTIVE.....*27*
EV_NOTACTIVE.....*27*
EV_STATUS.....*28*
EV_ACK_???.....*28*
EV_MESSAGE.....*30*
EV_CLEAR_???.....*30*
EV_DIVACK_???.....*31*
EV_RADIO_PER.....*32*
EV_NUM_INF.....*34*
EV_OPER_COND.....*35*
EV_NET_INF.....*36*
EV_RADIO_SET.....*37*
EV_RPROTO_INFO.....*38*
EV_RADIO_TSTRES.....*39*
CHAPTER V: MODEM MESSAGES.....41
M27_MakeModemCall.....*41*
M27_SendModemData.....*42*
CHAPTER VI - MODEM EVENTS.....43
EV_SETUP_???.....*43*
EV_INCOM_VMC.....*44*
EV_INCOM_EVMC.....*45*
EV_RECEIVE_???.....*45*
EV_RECEIVE_MD.....*47*

If you install or use this product, you will be deemed to be bound by the terms of the SOFTWARE license shown below.

LICENSE AGREEMENT

1. **License.** In consideration of payment of the license fee, wich is included in the price of the product, the License (you) is granted by the Licensor a non-exclusive right to use the SOFTWARE and associated documentation. No ownership rights to the SOFTWARE or its Documentation are transfered from Licensor to you.

2. **Term.** This license Agreement is effective until terminated. You may terminate this Agreement by destroying the SOFTWARE and documentation. You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. The Licensor may terminate this Agreement without notice if you violate any terms or conditions of the Agreement. In the event of termination of the Agreement, provisions relating to Author's disclaimers of warranties, limitation of liability, remedies, or damages and Author's propietary rights shall survive.

3. **Object Code.** You shall not reverse compile or otherwise reverse engineer the SOFTWARE.

4. **Limited Warranty.** This Software have commercial Royalty-Free License and is sold as-is.

INTRODUCTION

Please note that this User's Guide is based on the DLL product format, however all concepts are the same for the OCX product format.

FOR A BETTER COMPREHENSION:

We assume that you are familiarized with or you can obtain help on this topics if necessary:

- ✓ General knowledge in a lenguaje that uses DLLs.
- ✓ Dynamic Link Libraries and callback functions use.
- ✓ MAP27 compliant devices.

DOCUMENTATION COVENTIONS

The following conventions are used in this document:

To indicate a particular key, the name of the key is given in capitals. For example, press the ENTER key.

Sometimes you need to hold down one key on the computer while pressing another key. This is indicated by giving the name of the first key, then a plus "+", then the name of the second key. For example, "Ctrl+C" means "press the key labeled "CTRL" or "Ctrl"(i.e., the

"control"·key) and, while continuing to hold it down, press the "C" key. Multiple-key combinations that generate a single character are shown in angle brackets, like this: <Ctrl+C>

RU is the radio unit which communicate to the DTE using MAP27. DTE is a PC or device that is connected to the RU and controlled using MAP27.

DESCRIPTION

M27W32 is a dynamic link library that implements MAP27 protocol and allows you to incorporate it into your software.

This library is really easy to use because expose functions to execute actions and only one function to process the received messages, this is why we achieve a easy and clear use. This functions and messages had been developed using words as the specified by the protocol to allows a natural and direct understanding.

We recommend read the examples included to achieve a better product comprehension.

This library is protected by a hardware key and its drivers must be installed before start using it(see protection key chapter).

INSTALLATION NOTES

GENERAL

Product's files will be installed in the selected directory (default: c:\program files\MAP27W32) and no program group in the Start menú will be created.

In the installation directory the following sub-directories will be present:

- DLL: Files for the DLL format library.
- DOC: Documentation.
- OCX: Files for the OCX format library.
- Hardkey: hardware protection key (Do not apply for the UNLIMITED Licence).
- Samples: DLL and OCX sample applications.

OCX FORMAT

Remember that in order to use the OCX library format, you must register the component using the regsvr32.exe application, packaged in Windows OS.

CHAPTER I – LIBRARY USAGE

USING THE LIBRARY INTO YOUR SYSTEM

Incorporating the library into your system is easy and is detailed below.

- ***C++:*** Include in your project map27w32.lib and map27w32.h files, which contains all the necessary declarations. Besides you must copy the DLL file in Windows directory or your project working directory.

- ***Visual Basic:*** Include in your project M27W32.BAS file which contains all the declarations necessary. Besides you must copy the DLL file in Windows directory or your working directory, which in Visual Basic generally is not the project directory.

IMPORTANT
We recommend reading the source code included as example because are working applications and a unique experience source.

FUNCTION CALLBACK

Messaging from RU processing is made trough a user defined callback function. The function sintax must be:

C++:

DWORD WINAPI MyFunctionName(BYTE Type,BYTE Prefix, WORD Ident, BYTE Status,BYTE DataLen);

Visual Basic:

Function MyCallBack(ByVal pType As byte, ByVal pPrefix As Byte, ByVal pIdent As Long, ByVal pStatus As byte, ByVal pDatalen As Byte) As Integer

where:

Type Incoming message type. Message types are defined with constants like EV_xxxxxxx. For more information see corresponding chapter.

Prefix: Calling RU prefix.

Ident: Calling RU identification.

Status: Status variable.

Datalen: Length, in bytes, of data received in the reception buffer. This data could be retrieve using GetData function.

Remember that this function must return the library control because it could break the protocol normal working.

INITIALIZING

Before you use any function in this library you must initialize it in order to prepare the internal buffers and start protocol working. The initialization procedure is:

1. M27_SetCallBack to assign the message processing function.
2. M27_SetRS232Config to set the serial port to use.
3. M27_InitialiseMap27 to start protocol operation.

After library initializing the library you must wait a EV_ACTIVE event that indicates RU-DTE link.

WARNING
Visual Basic must know callback use limitations because this functions presents some considerations in step by step running mode.

CLEANING UP

Before exiting you application, or if you need to stop the internal protocol machinery, you must use M27_CleanupMap27 function.

WARNING
Avoiding the use of this function could generate unknown errors in your system.

CHAPTER II - GENERAL FUNCTIONS

M27_GetLibVersion

DESCRIPTION:

Returns internal library version.

DECLARATION:

```
DWORD MAP27W32_API  M27_GetLibVersion(VOID);
```

PARAMETERS:

None.

RETURNS:

Library internal version. Now 103.

M27_InitialiseMap27

DESCRIPTION:

Start protocol working. Before using any library function you must initialize it, in order to prepare internal buffers and to start protocol working. The initialization procedure is:

1. M27_SetCallBack to assign the messaging processing function.

2. M27_SetRS232Config to sets serial port to use.
3. M27_InitialiseMap27 to start protocol working.

DECLARATION:

WORD MAP27W32_API M27_InitialiseMap27(VOID)

PARAMETERS:

None.

RETURNS:

M27_SUCESS on correct initialization. M27_FAILPORT on a serial port error or M27_ERROR on other errors.

M27_CLEANUPMAP27

DESCRIPTION:

Ends protocol execution.

DECLARATION:

WORD MAP27W32_API M27_CleanupMap27(VOID);

PARAMETERS:

None.

RETURNS:

M27_SUCESS on success. M27_ERROR on error.

M27_SetRS232Config

DESCRIPTION:

Sets serial port parameters. This parameters depends on the MAP27 devices to use.

DECLARATION:

```
WORD MAP27W32_API M27_SetRS232Config(LPCSTR ComName,
        int BaudRate, int Parity, int Stop, int Data);
```

PARAMETERS:

- ✓ *ComName:* Serial port name where the RU is connected. i.e.: COM2.
- ✓ *BaudRate:* Communication speed in bauds. Default: 9600 bps.
- ✓ *Parity:* Parity to use. Default: 0 (Ninguna).
- ✓ *Stop:* Stop Bits. Default: 1 bit.
- ✓ *Data:* Data Bits. Default: 8 bits.

RETURNS:

M27_SUCESS if success. M27_FAILSETTINGS on error.

M27_GetRS232Config

DESCRIPTION:

Return serial port settings.

DECLARATION:

```
WORD MAP27W32_API M27_GetRS232Config(LPSTR ComName,
        int *BaudRate, int *Parity, int *Stop, int *Data)
```

PARAMETERS:

- ✓ *ComName:* Variable to store the selected serial port.
- ✓ *BaudRate:* Variable to store port speed.
- ✓ *Parity:* Variable to store port parity.
- ✓ *Stop:* Variable to store port stop bits.
- ✓ *Data:* Variable to store port data bits.

RETURNS:

M27_SUCESS on success. M27_FAILSETTINGS on error.

M27_SetCallBack

DESCRIPTION:

Sets message processing function.

DECLARATION:

VOID MAP27W32_API M27_SetCallBack(MAP27_CALLBACK pFnc);

PARAMETERS:

- ✓ *pFnc: P*ointer to the function that will process messages from RU.

RETURNS:

None.

M27_GetData

DESCRIPTION:

Returns reception buffer data. This data vary according to the message type. This information is detailed in each message description in next chapters.

DECLARATION:

VOID MAP27W32_API M27_GetData(LPSTR pData)

PARAMETERS:

- ✓ *pData:* Variable to store reception buffer data.

RETURNS:

None.

M27_SetCodeInfo

DESCRIPTION:

Sets transmission coding for short and long messages.

DECLARATION:

BYTE MAP27W32_API M27_SetCodeInfo(BYTE Coding)

PARAMETERS:

- ✓ *Coding:* Coding type:

  MPT1343_BCD_RP: BCD Radio Path Coding.

  MPT1343_CCITT_RP: CCITT Alphabet No 2 (TELEX)

  MPT1343_BINARY: Binary

  MPT1343_BCD: BCD

  MPT1343_CCITT: Alphabet No 2 (TELEX)

  MPT1343_CCITT_5: CCITT Alphabet No 5 (7 bit ASCII)

MPT1343_PC8: 8 Bit PC character set.

RETURNS:

M27_SUCCESS

CHAPTER III – DTE-RU MESSAGES

M27_SendStatusMessage

DESCRIPTION:

Sends an status message. Status messages are numeric in the range from 0 to 31.

DECLARATION:

WORD MAP27W32_API SendStatusMessage(BYTE prefix, WORD ident, BYTE status)

PARAMETERS:

- ✓ *Prefix:* Called party prefix. i.e..: 2.
- ✓ *ident;* Called party identification. i.e.: 31.
- ✓ *status:* Status value to send(value between 0 and 31).

RETURNS:

M27_SUCCESS if the command was accepted by protocol internal layers. M27_ERROR on error. Remember that M27_SUCCESS not means that the other radio received the command. After sending the command you must wait an ACK to consider the successful reception.

M27_SendMessage

DESCRIPTION:

Sends a short or long message. The maximum data length is 88 bytes. You must receive an EV_ACK event if transmission was successful. If the message was segmented you will receive as many EV_QUE as the central informs.

DECLARATION:

WORD MAP27W32_API M27_SendMessage(BYTE prefix, WORD ident, LPVOID data, BYTE dataLen)

PARAMETERS:

- ✓ *Prefix:* Called party prefix. i.e.: 2.
- ✓ *ident;* Called party identification. i.e.: 31.
- ✓ *Data:* Message data.
- ✓ *DataLen:* Message data length (in bytes).

RETURNS:

M27_SUCCESS command was received by protocol internal layers. M27_ERROR on error.

M27_ClearCall

DESCRIPTION:

Cancels a call request, a transaction or disconnects a modem/voice call.

DECLARATION:

WORD MAP27W32_API M27_ClearCall(BYTE prefix, WORD ident, BYTE type)

PARAMETERS:

- ✓ *prefix:* Called party prefix.
- ✓ *ident:* Called party identification.
- ✓ *Type:*

  CC_NORMAL: Disconnect voice or modem call, normal end.

  CC_ANY: Cancel any message transaction or normal call set-up attempt.

  CC_INCLUDE: Cancel include call set-up attempt.

  CC_DIVERSION: Abort diversion setting transaction.

  CC_NPDCALL: Cancel standard data call set-up attempt.

NOTE
Values could be combined, using an OR logic operation.

RETURNS:

M27_SUCCESS if successful. M27_ERROR on error

M27_RADIOPERSONALITY

DESCRIPTION:

Asks the radio unit for its personality and present operating conditions. (see EV_RADIO_PER).

DECLARATION:

```
WORD  MAP27W32_API  M27_RadioPersonality(VOID)
```

PARAMETERS:

None.

RETURNS:

M27_SUCCESS if command was accepted by the internal protocol layers. M27_ERROR on error.

M27_NUMBERINGINFO

DESCRIPTION:

Asks the radio unit for its numbering information(see EV_NUM_INF).

DECLARATION:

```
WORD  MAP27W32_API  M27_NumberingInfo(VOID)
```

PARAMETERS:

None.

RETURNS:

M27_SUCCESS if command was received by the internal protocol layers. M27_ERROR on error.

M27_STATUS

DESCRIPTION:

Asks RU working state.

DECLARATION:

WORD MAP27W32_API M27_Status(VOID)

PARAMETERS:

None.

RETURNS:

M27_SUCCESS if command was received by the internal protocol layers. M27_ERROR on error.

M27_OperatingConditions

DESCRIPTION:

Asks the radio unit for its operating conditions.

DECLARATION:

WORD MAP27W32_API M27_OperatingConditions(VOID)

PARAMETERS:

None.

RETURNS:

M27_SUCCESS. M27_ERROR on error.

M27_NetworkInfo

DESCRIPTION:

Asks the radio unit for its Network information.

DECLARATION:

WORD MAP27W32_API M27_NetworkInfo(VOID)

PARAMETERS:

None.

RETURNS:

M27_SUCCESS. M27_ERROR on error.

M27_RadioControl

DESCRIPTION:

DTE informs the radio unit of the user controlled actions needed in the radio path protocol.

DECLARATION:

WORD MAP27W32_API M27_RadioControl(BYTE ctrls)

PARAMETERS:

- ✓ *ctrls:* are:

  RC_OFFHOOK: Off-hook indication i.e. user ready for a modem or speech call.

  RC_TORQST: Transmit On request.

RETURNS:

M27_SUCCESS. M27_ERROR on error.

M27_SetNewVolume

DESCRIPTION:

The DTE may send this command to the radio unit to adjust audio levels.

DECLARATION:

WORD MAP27W32_API M27_SetNewVolume(BYTE vol, BYTE audiosource)

PARAMETERS:

- ✓ *vol:* Possible values are:

  SET_UP: Up.

  SET_DOWN: Down.

  SET_TO: set to value in range 0..15, '0' is the lowest audio volume.

  SET_PRESET: Set to preset value e.g. value defined to emergency traffic(optional).

  SET_MANUAL: Reset to manual control(optional).

- ✓ *audiosource:* Possible values are:

  SRC_SPEECH: Normal speech audio path.

  SRC_ATONES: Alert tones.

  SRC_MODEM: Modem audio path (audio level for a modem).

RETURNS:

M27_SUCCESS. M27_ERROR on error.

M27_DialledStrings

DESCRIPTION:

The DTE may send this command to the radio unit to initiate actions according to MPT1343. Call set-ups are not supported by using this message.

DECLARATION:

WORD MAP27W32_API M27_DialledStrings(LPVOID data, BYTE len)

PARAMETERS:

- ✓ *data:* Characters used for dialing control actions.
- ✓ *len:* data length in bytes.

RETURNS:

M27_SUCCESS. M27_ERROR on error.

M27_RADIOTEST

DESCRIPTION:

The DTE may send this command to the radio unit to initiate manufacturer specific radio testing.

DECLARATION:

WORD MAP27W32_API M27_RadioTest(LPVOID data, BYTE len)

PARAMETERS:

- ✓ *data:* Test controls.
- ✓ *len:* data length in bytes.

RETURNS:

M27_SUCCESS. M27_ERROR on error.

M27_MakeRadioSpeechCall

DESCRIPTION:

This command is sent by the DTE to the radio unit and contains the information needed to set-up a voice call.

DECLARACION:

WORD MAP27W32_API M27_MakeRadioSpeechCall(BYTE prefix, WORD ident, BYTE priority, BOOL include, BOOL group)

PARAMETERS:

- ✓ *Prefix:* Called party prefix. i.e.: 2.
- ✓ *ident;* Called party Identification. i.e.: 31.
- ✓ *priority:* Possible values are:

  M27_NONP: default. Normal priority.

  M27_HIGH: High priority.

  M27_EMERGENCY: Emergency call.(may be combined with Normal o High).

- ✓ *include:* TRUE indicates include call.
- ✓ *group:* TRUE indicates group call, called users are not allowed to reply.

RETURNS:

M27_SUCCESS. M27_ERROR on error.

CHAPTER IV – EVENTS RU-DTE

EV_ACTIVE

DESCRIPTION

Indicates RU-DTE linkage. After this event was received the system was able to execute the library commands.

PARAMETERS

- ✓ Type: EV_ACTIVE
- ✓ Prefix: 0
- ✓ Ident: 0
- ✓ Status: 0
- ✓ Datalen: 0

EV_NOTACTIVE

DESCRIPTION

Indicates a RU-DTE linkage lost, you may check the possible cause.

PARAMETERS

- ✓ Type: EV_NOTACTIVE
- ✓ Prefix: 0
- ✓ Ident: 0
- ✓ Status: 0

- ✓ Datalen: 0

EV_STATUS

DESCRIPTION

Indicates that a status message was received.

PARAMETERS

- ✓ Type: EV_STATUS
- ✓ Prefix: Calling party prefix. i.e.: 2
- ✓ Ident: Calling party identification. i.e.: 31
- ✓ Status: Status number. Range 0 a 31.
- ✓ Datalen: Data length. Always 1, because the status value is the first byte in the reception buffer.

EV_ACK_???

DESCRIPTION

This event is generated by the RU to the DTE and either acknowledge that the called party radio or line unit has received the previous command or indicates the reason for an unsuccessful attempt.

PARAMETERS

- ✓ Type:

  EV_ACK_POS: Successful transaction.

  EV_ACK_QUE: Called unit engaged, waiting for signaling. See cause.

  EV_ACK_NEG: Transaction aborted. See cause.

- ✓ Prefix: 0.
- ✓ Ident: 0.

- ✓ Status: Reason for sending this message.
- ✓ Datalen: 1. First reception buffer is the status value.

ACK_POS causes

Cause	Description
AC_OK	Successful transaction

ACK_QUE causes

Cause	Description
AC_BUSY	System busy, waiting for signalling
AC_WAIT	Called unit engaged, wait for signalling
AC_DIVSEND	Called unit's calls are diverted and radio unit tries to send message to the diversion address.
AC_TRIESST	TSC does not support MSG, the radio tries to use multiple SST.

ACK_NEG causes

Causa	Description
AC_ABORTED	Transaction aborted
AC_INVALID	Invalid call, message rejected
AC_OVERLOAD	System or called unit overload, message rejected
AC_ABANDONED	Called radio out of reach or transaction abandoned
AC_ENGAGED	Called unit engaged or does not wish to accept message
AC_DIVERTED	Called unit's calls are diverted
AC_DIVTOGROUP	Called unit's calls are diverted to a group address
AC_DIVNOTAVAL	Called unit's calls are diverted, but the diversion address is not available
AC_MSTNOTSUPP	TSC does not support MSG, transaction aborted

EV_MESSAGE

DESCRIPTION

Indicates data message reception. The message could be short (up to 22 bytes) or long (up to 88 bytes).

PARAMETERS

- ✓ Type: EV_MESSAGE
- ✓ Prefix: Called party prefix. i.e.: 2
- ✓ Ident: Called party identification. i.e.: 31
- ✓ Status: 0.
- ✓ Datalen: Reception buffer length.

EV_CLEAR_???

DESCRIPTION

This event is generated by the radio unit to the DTE to indicate a cancelled transaction or a disconnected call.

PARAMETERS

- ✓ Type:

  EV_CLEAR_NOR: Final normal.

  EV_CLEAR_ABN: Final abnormal. (see cause).
- ✓ Prefix: Called party prefix. i.e.: 2
- ✓ Ident: Called party identification. i.e.: 31
- ✓ Status: Cancellation or disconnection reason.
- ✓ Datalen: 1. Reception buffer data length.

CLEAR_ABN causes

Cause	Description
CLC_UNKNOWN	Not specified, all message transactions, call set-ups or calls are cancelled or disconnected
CLC_RADIOGEN	Radio generated clear e.g. radio path protocol time-out or on-hook on the radio set.
CLC_NOTAVAL	Service not available (Radio unit not in radio contact)
CLC_TOOLONG	Transmission or message too long, call disconnected or message rejected
CLC_DISCABN	Voice or modem call disconnecteed, abnormal end

CLEAR_NOR causes

Cause	Description
CLC_DISNOR	Voice or modem call disconnected, abnormal end

EV_DIVACK_???

DESCRIPTION

This event is generated by the RU to the DTE to indicate the progress, acceptance or rejection of the message setting or cancellation a diversion.

PARAMETERS

- ✓ Type:

  EV_DIVACK_POS: Call diversion or diversion cancellation accepted.

  EV_DIVACK_NEG: Transaction rejected(see cause).

- ✓ Prefix: Diversion target prefix. i.e.: 2
- ✓ Ident: Diversion target identification. Por ej.: 31

- ✓ Status: Reason for sending this message.
- ✓ Datalen: 1. Data length in reception buffer.

DIVACK_POS cause

Cause	Description
DAC_OK	Call diversion or diversion cancellation accepted

DIVACK_NEG cause

Cause	Description
DAC_INVALID	Invalid diversion or TSC does not accept diversion, request rejected
DAC_OVERLOAD	System overload, request rejected
DAC_ABANDONED	Transaction abandoned

EV_RADIO_PER

DESCRIPTION

This event is generated by RU as the response to a radio personality interrogation(see radio personality command).

PARAMETERS

- ✓ Type: EV_RADIO_PER
- ✓ Prefix: RU prefix.
- ✓ Ident: RU identification.
- ✓ Status: 0.
- ✓ Datalen: 8 bytes. Reception buffer data length.

Received data will be:

Byte	Description
1	Manufacturer's code(8 bits binary number)
2	Model number in the bit positions 8 to 5 and two most significant bits 18 and 17 of the serial number in the bit positions 2 and 1
3	Serial number bits 16 to 9
4	Serial number bits 8 to 1
5	Supported facility(group A)
6	Supported facility(group B)
7	Supported facility(group C)
8	Supported SST and MST coding types.

Facilities Group A

Bit	Description
8	Voice calls support
7	Modem calls support
6	Status messages support
5	SST messages support
4	MST messages support
3	Automatic call set-up to diversion address support
2	Call-back support
1	reserved

Facilities Group B

Bit	Description
8…1	Reserved for further extensions, set to '0'

Facilities Group C

Bit	Description
8…1	Space for customization

Supported Codings

Bit	Description
8	MPT1343 BCD presentation as ASCII characters support
7	MPT1343 CCITT Alphabet No 2 (Telex) presentation as ASCII support
6	MPT1343 binary presentation support
5	MPT1343 BCD presentation support
4	MPT1343: CCITT Alphabet No 2 (Telex). Characters are presented as Telex characters
3	MPT1343: CCITT Alphabet No 5(7 bits ASCII). Number of ASCII characters is in the range of 1..25
2	MPT1343: eight bit characters according to PC character set. The recommended character set number is 437
1	Reserved

EV_NUM_INF

DESCRIPTION

This event is generated by the RU to the DTE either unsolicited or as the response to a numbering information interrogation

PARAMETERS

- ✓ Type: EV_NUM_INF
- ✓ Prefix: RU prefix.
- ✓ Ident: RU identification.

- ✓ Status: 0.
- ✓ Datalen: Reception buffer length.

> **NOTE**
> For details on message information please refer to protocol specification.

EV_OPER_COND

DESCRIPTION

The RU may generate this message unsolicited or as a response to the operating condition network information interrogation messages and radio control command message.

PARAMETERS

- ✓ Type: EV_RADIO_COND
- ✓ Prefix: 0.
- ✓ Ident: 0.
- ✓ Status: 1 Byte determining the operating conditions.
- ✓ Datalen: Reception buffer data length.

Reception buffer data:

Byte	Description
1	Operating conditions
2	Received field strength
3	Maximum call duration time

Operating conditions:

Value	Description
OP_CONTACT	Radio unit contact
OP_TRANSMITING	Radio unit transmitting

Field strength:

Value	Description
0	Dummy valu, field strength not available
1	Lowest field strength
255	Highest field strength
Otros	In between field strength

Maximum call duration time:

Value	Description
0	Reserved
1	5 min
9	13 min
10	10 sec
254	254 sec
255	Call duration timer infinite or not supported

EV_NET_INF

DESCRIPTION

RU may generate this event solicited or as a response to the network information interrogation command.

PARAMETERS

- ✓ Type: EV_NET_INF
- ✓ Prefix: 0.
- ✓ Ident: 0.
- ✓ Status: 0.
- ✓ Datalen: Reception buffer length.

Reception buffer:

Byte	Description
1	Radio channel number, the highest order bit in the bit position 2
2	Radio channel number, the lowest order bit in the bit position 1
3	System identity code, the highest order bit in the bit position 7
4	System identity code, the lowest order bit in the bit position 1

EV_RADIO_SET

DESCRIPTION

The RU may generate this message as an acknowledgement to the radio management message or as the response to de radio control command to indicate the activated settings of the RU

PARAMETROS

- ✓ Type: EV_RADIO_SET
- ✓ Prefix: 0.
- ✓ Ident: 0.
- ✓ Status: Radio settings.
- ✓ Datalen: Reception buffer data length.

Radio Settings:

Byte	Descripción
1	Activated facilities(see below)
2	Reserved for further extensions, set to '0'
3	Spare for customization

Activated facilities:

Value	Description
RS_VOICE	RU accepts incoming voice calls
RS_MODEM	RU accepts incoming modem calls
RS_STATUS	RU accepts incoming status messages
RS_SST	RU accepts incoming SST messages
RS_MST	RU accepts incoming MST messages
RS_DIV	RU does automatically set up calls to a diversion address
RS_CALLBACK	Call back logging
RS_RESERVED	Reserved for further extension, set to '0'

EV_RPROTO_INFO

DESCRIPTION

The DTE or RU may generate this event unsolicited or as a response to indicate a protocol dificulty.

PARAMETERS

- ✓ Type: EV_RPROTO_INFO
- ✓ Prefix: 0.

- ✓ Ident: 0.
- ✓ Status: Dificultad.
- ✓ Datalen: Reception buffer data length.

Reception buffer:

Byte	Description
1	Reason to send this message

Reasons:

Value	Description
1	Unrecognised message
2	facility or addressing not supported
3	Protocol state mimatch detected i.e. received message not compatible or allowed at the present state (optional)
4	Message coding not supported
>=128	Spare for customization
Otros	Reserved

EV_RADIO_TSTRES

DESCRIPCION

The RU may generate this event to the DTE as a response to the manufacturer specific rado testing.

PARAMETROS

- ✓ Type: EV_RADIO_TSTRES
- ✓ Prefix: 0.

- ✓ Ident: 0.
- ✓ Status: 0.
- ✓ Datalen: Reception buffer data length.

NOTE
Reception buffer may contain one or more octets specified by manufacturer.

CHAPTER V: MODEM MESSAGES

In this chapter are explained modem functions. Note that the implementation of this functions depends on the RU manufacturer' s specification.

Moreover, functions as RadioControl that are not included in this chapter (because are not modem only function) but will be use in this type of communication according to manufacturer's specification.

M27_MakeModemCall

DESCRIPTION:

This message is sent by the DTE to the RU and contains all the information needed to set-up a voice or modem call.

DECLARATION:

```
WORD  MAP27W32_API  M27_MakeModemCall(BYTE prefix, WORD
                                      ident, BYTE priority)
```

PARAMETERS:

- ✓ *Prefix:* Called party prefix. i.e.: 2.
- ✓ *ident;* Called party identification. i.e.: 31.
- ✓ *priority:*

  M27_NONP: default. Normal Priority.

  M27_HIGH: High priority.

  M27_EMERGENCY: Emergency call. (This value may be combined with NONP or HIGH using OR operator).

RETURNS:

M27_SUCCESS . M27_ERROR on error.

M27_SENDMODEMDATA

DESCRIPTION:

Send user data to the RU or modem during a modem call. The message is optional. Usage and user data format of this message is implementation specific.

DECLARATION:

```
WORD  MAP27W32_API  M27_SendModemData(LPVOID data,
                                      WORD len)
```

PARAMETERS:

- ✓ *data:* Free format user data.
- ✓ *Len:* Data length.

RETURNS:

M27_SUCCESS . M27_ERROR on error.

CHAPTER VI - MODEM EVENTS

EV_SETUP_???

DESCRIPTION

This event is generated by the RU to indicate the progress, acceptance o r rejection of a voice or modem call.

PARAMETERS

- ✓ Type:

  EV_SETUP_POS: Connected.

  EV_SETUP_QUE: Waiting. See cause.

  EV_SETUP_NEG: Aborted. See cause.

- ✓ *Prefix:* Called party prefix.
- ✓ *ident;* Called party identification.
- ✓ Status: Cause.
- ✓ Datalen: 1. First byte in the reception buffer is the cause.

SETUP_POS cause

Cause	Description
SC_OK	Connected

SETUP_QUE cause

Cause	Description

SC_ALERTING	Called unit alerting
SC_BUSY	System busy, wait for signalling
SC_WAIT	Called unit engaged, wait for signalling
SC_PROCEEDING	Emergency call is proceeding, wait for signalling
SC_DIVSEND	Called unit's calls are diverted and RU tries to setup call to the diversion address
SC_TRIESDIV	Called unit's calls are diverted to a group and radio unit tries to set-up call to the diversion address

SETUP_NEG cause

Cause	Description
SC_ABORTED	Call setup aborted
SC_REJECTED	Invalid call, call setup rejected
SC_OVERLOAD	System or called unit overload, call setup rejected
SC_ABANDONED	Called radio out of reach or call setup abandoned
SC_ENGAGED	Called unit engaged or user does not wish to accept the call
SC_DIVERTED	Called unit's calls are diverted
SC_DIVTOGROUP	Called unit's calls are diverted to a group address
SC_DIVNOTAVAL	Called unit's calls are diverted, but the diversion address is not available
SC_CALLBACK	Called unit has acceptd the call for call-back

EV_INCOM_VMC

DESCRIPTION

Indicates and incoming voice or modem call.

PARAMETERS

- ✓ Type: EV_INCOM_VMC
- ✓ Prefix: Calling party prefix.
- ✓ Ident: Calling party identification.
- ✓ Status: Byte with call details according to MAP27 protocol.
- ✓ Datalen: Allowed range 0..44 bits in 5 and half octets(Valid only with customized services).

EV_INCOM_EVMC

DESCRIPTION

Indicates an incoming emergency voice or modem call.

PARAMETERS

- ✓ Type: EV_INCOM_VMC
- ✓ Prefix: Calling party prefix.
- ✓ Ident: Calling party identification.
- ✓ Status: Byte with call details according to MAP27 protocol.
- ✓ Datalen: Allowed range 0..44 bits in 5 and half octets(Valid only with customized services).

EV_RECEIVE_???

DESCRIPTION

RU generates this event to indicate progress, acceptance or rejection of a voice or modem call.

PARAMETERS

- ✓ Type:

  EV_RECEIVE_POS: Connected.

  EV_RECEIVE_WAR: Waiting. See cause.

  EV_RECEIVE_CNC: Aborted. See cause.

- ✓ Prefix: Calling party prefix.
- ✓ Ident: Calling party identification.
- ✓ Status: Cause.
- ✓ Datalen: 1. First byte is the cause.

Cause RECEIVE_POS

Cause	Description
RC_OK	Connected

RECEIVE_WAR cause

Cause	Description
RC_DISABLED	Transmision disabled
RC_BUSY	System busy, wait for signalling
RC_CLEARDOWN	Clear down timer warning(optional)

RECEIVE_CNC cause

Cause	Description
RC_REJECTED	Call setup rejected
RC_CALLBACK	Radio has accepted the call for call-back

EV_RECEIVE_MD

DESCRIPTION

This event carries user data and is generated by the RU or modem during a modem call. Usage and user data format of this message is application specific.

PARAMETERS

- ✓ Type: EV_RECEIVE_MD
- ✓ Prefix: 0.
- ✓ Ident: 0.
- ✓ Status: 0.
- ✓ Datalen: Reception buffer length.