READBEST

RDS Encoder

Version 1.5

Implementation Manual

2020

Pira.cz

Table of Content

1. Product Overview

1.1. Product Features

- Fully dynamic RDS encoder written in C, for adding RDS functionality into various FM broadcast products
- The RDS encoder's internal structure is based on UECP 7 specification
- Enhanced by a set of ASCII commands and additional features
- Multiple communication ports internally supported
- The set of UECP and ASCII commands ensures support in all broadcast automation systems, in order to send actual song/program information, announcements, traffic information etc.
- Advanced text functions (Dynamic/scrolling PS, word parsing)
- Supports RDS Spy – remote output monitoring in real time (only in version 1.5 or later)
- Fully supported by Magic RDS 4 control software
- Demo application available as a part of the Magic RDS 4

1.2. Recommended References

[1] UECP Specification (SPB 490), RDS Forum, Geneva

1.3. Technical Characteristics

Parameter	**Value**
Number of Communication Ports	2 *
Number of Data sets	4 *
Number of Program services inside each Data set	9 *
Max. number of Radiotexts inside each Data set	10 *
Max. length of Dynamic/scrolling PS loop	128 characters *
Max. length of single set of AF lists for main Program service	512 bytes *
Max. length of single set of AF lists for EON	128 bytes *
Max. length of Free-format group buffer	8 groups per each group type *
RAM requirements for RDS content	18 kB (typical)
Non-volatile memory requirements	12 kB (typical)

** Note: Predefined value. Parameter configurable by the software engineer.*

1.4.RDS Services Directly Supported

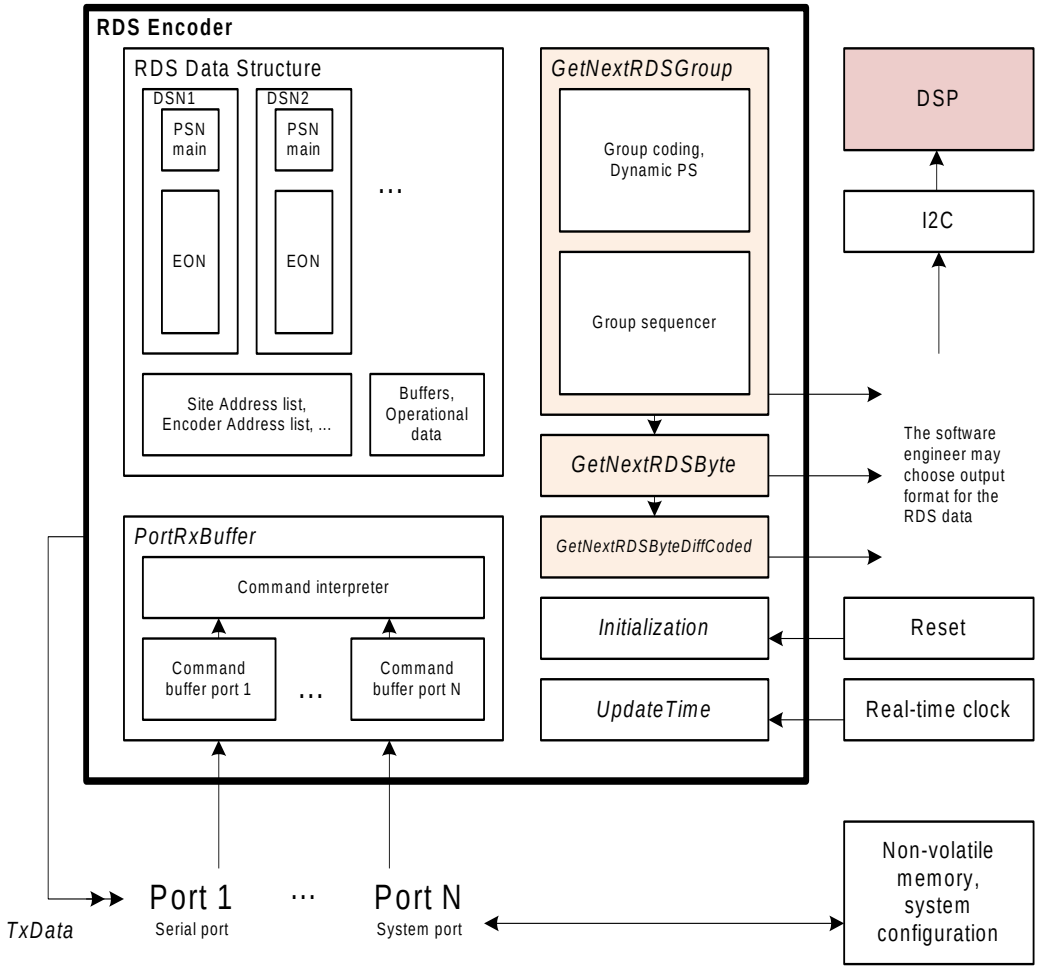
RDS Service		Description
PI	Programme identification	The Programme Identification (PI) is a code enabling the receiver to distinguish between audio programme content. The most important application of the PI code is to enable the receiver in the event of bad reception, to switch automatically from the currently used frequency to an alternative frequency - the criterion for the changeover to the new frequency would be the presence of a better signal having the same Programme Identification code. It follows therefore that the PI must be allocated in such a way that it uniquely distinguishes each audio programmme content from all others in the same area. The PI is it not intended for direct display.
PS	Programme service name	This is the label of the programme service consisting of eight alphanumeric characters, which is displayed by RDS receivers in order to inform the listener what programme service is being broadcast by the station to which the receiver is tuned. An example for a name is 'Radio 21'. **Dynamic/scrolling PS supported as well** (function prohibited by the RDS standard).
DI	Decoder identification	These bits indicate which possible operating modes are appropriate for use with the broadcast audio.
TP	Traffic-programme identification	This is a flag to indicate that the tuned programme carries traffic announcements. The TP flag must only be set on programmes which dynamically switch on the TA identification during traffic announcements. The signal shall be taken into account during automatic search tuning.
TA	Traffic-announcement identification	This is an on/off switching signal to indicate when a traffic announcement is on air.
MS	Music/speech switch	This is a two-state signal to provide information on whether music or speech is being broadcast.
PTY	Programme type	This is an identification number to be transmitted with each programme item and which is intended to specify the current Programme Type within 31 possibilities. This code could be used for search tuning.
PTYN	Program type name	The PTYN feature is used to further describe current PTY. PTYN permits the display of a more specific PTY description that the broadcaster can freely decide (e.g. PTY=4: Sport and PTYN: Football). The PTYN is not intended to change the default eight characters of PTY which will be used during search or wait modes, but only to show in detail the program type once tuned to a program. If the broadcaster is satisfied with a default PTY name, it is not necessary to use additional data capacity for PTYN.
RT	Radiotext	This refers to 64-character long text transmissions addressed to receivers, which would be equipped with suitable display facilities.
AF	List of alternative frequencies	The list(s) of alternative frequencies give information on the various transmitters broadcasting the same programme in the same or adjacent reception areas, and enable receivers equipped with a memory to store

		the list(s), to reduce the time for switching to another transmitter. This facility is particularly useful in the case of car and portable radios.
EON	Enhanced information on other networks	This feature can be used to update the information stored in a receiver about programme services other than the one received. Alternative frequencies, the PS name, Traffic Programme and Traffic Announcement identification as well as Programme Type can be transmitted for each other service. The relation to the corresponding programme is established by means of the relevant Programme Identification (PI).
LI	Linkage information	Linkage information provides the means by which several program services, each characterized by its own PI code, may be treated by a receiver as a single service during times a common program is carried.
LIC	Language identification code	To enable a broadcaster to indicate the spoken language he is currently transmitting.
ECC	Extended country code	It helps the receiver to recognise the country in cooperation with the PI code. The first most significant bits of the PI code carry the RDS country code. The four bit coding structure only permits the definition of 15 different codes, 1 to F (hex). Since there are much more countries to be identified, some countries have to share the same code which does not permit unique identification. The ECC byte determines the country unambigouesly.
CT	Clock time and date	Time and date codes shall use Coordinated Universal Time (UTC) and Modified Julian Day (MJD). These codes are intended to update a free running clock in a receiver. The listener, however, will not use this information directly and the conversion to local time and date will be made in the receiver's circuitry.
PIN	Program-item number	The code should enable receivers and recorders designed to make use of this feature to respond to the particular program item(s) that the user has preselected. Use is made of the scheduled program time, to which is added the day of the month. The transmitted Program Item Number code will be the scheduled broadcast start time and day of month as published by the broadcaster.
TDC	Transparent data channels	The transparent data channels consist of 32 channels which may be used to send any type of data.
IH	In-house applications	This refers to data to be decoded only by the operator. Some examples noted are identification of transmission origin, remote switching of networks and paging of staff. The applications of coding may be decided by each operator itself.
EWS	Emergency Warning System	The EWS feature is intended to provide for the coding of warning messages. These messages will be broadcast only in cases of emergency and will only be evaluated by special receivers.
TMC	Traffic Message Channel	This feature is intended to be used for the coded transmission of traffic information.
RAW	Free-format Groups	This feature may be used for testing purposes, ODA applications or software emulation of any service which is not supported directly.

2. Software Implementation

2.1. The Encoder's Software Model

Following diagram shows the encoder's internal logical structure and typical implementation of the encoder in parent application.



2.2. Programming Considerations

Note: This section applies to full source distribution only.

- Absolute independence on a platform/OS (no platform-specific operations included)
- No infinite loop, no priority levels, no waiting for user inputs or I/O operations
- Only a set of global variables and a set of functions
- The parent application is fully responsible for calling the functions anytime the events specified occur
- Multi-threaded processing of the RDS encoder's functions is not allowed – the parent application must ensure that no more than one RDS encoder's function is running at any time.

2.3. List of Files

Note: This section applies to full source distribution only.

File	Description
main.c	This file is not a part of the RDS encoder. It is an example of parent application instead. It compiles under Windows.
simple_tcp_server.c	This file is not a part of the RDS encoder. Required by the main.c.
rdsdemo.exe	Executable Demo Windows application.
rdsenc.c	**The encoder's main file.**
rdsacmd.c	ASCII command interpreter.
rdsuecp.c	UECP command interpreter.
rdsgrpc.c	RDS group coding.
rdsconv.c	Conversion/CRC routines.
**.h*	The encoder's header files. At least the rdsenc.h file must be included in your project (hereinafter referred to as *parent application*).
ini.cfg	An example of configuration commands, incl. demo of PS, PTY, RT set, Dynamic PS, EON, TMC, PTYN, AF, ECC, LIC, Group sequence, Free group, etc.

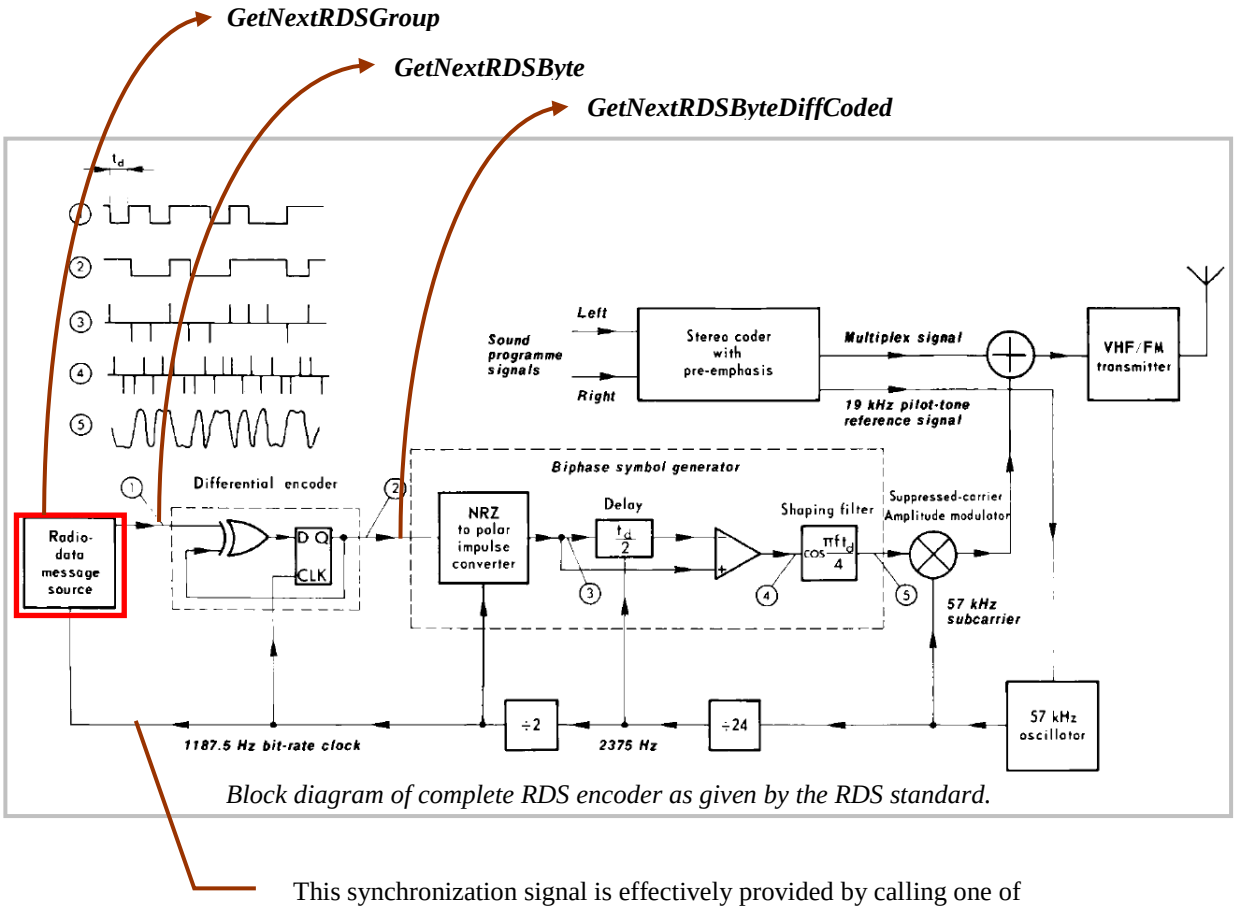
2.4. List of Functions to be Called

Note: This section applies to full source distribution only.

2.4.1. Functions for getting output RDS data

It is not allowed to call different output functions. The engineer must decide for one of these functions in advance, especially in dependence of the RDS modulation method requirements.

Function	Return value format, description
GetNextRDSGroup	Puts next RDS group to an array of integers given by pointer as the parameter. Valid RDS data are contained within the array of four integers. The array item [0] contains block A, the array item [1] contains block B, etc. Each block occupies bits 15 to 0 in the integer. CRC and offset word is not included. Direct call to this function is especially useful for debug.
GetNextRDSByte	Returns next RDS byte (prior to differential encoding). Valid RDS data are contained within bits 7 to 0. The bit 7 needs to be transmitted first. CRC and offset word is included.
GetNextRDSByteDiffCoded	Returns next RDS byte (differentially encoded). Valid RDS data are contained within bits 7 to 0. The bit 7 needs to be transmitted first. CRC and offset word is included.



Block diagram of complete RDS encoder as given by the RDS standard.

This synchronization signal is effectively provided by calling one of the functions above any time the next data is required.

2.4.2. Other functions

Function	Description
PortRxBuffer	The parent application signalizes by calling of this function that there is at least 1 new byte waiting in specified port input buffer. Reads the data from the port buffer and processes the UECP, ASCII or system commands. Calls the *TxData* function if any response has to be sent. Returns number of bytes **unread** (ignored) from the port buffer. If data flow control is not used, the return value may be ignored as entire data is always read.
Initialize	To be called on the equipment's power up or reset. As the parameter, pointer to TxData callback function is passed.
UpdateTime	Updates the RDS encoder's internal real-time clock. Can be called any time, recommended calling is after power-up and then once per hour. The time information passed is a UTC time. The RDS encoder's internal clock is running and kept accurate by calling the *GetNextRDS...* function because average frequency of calling is constant and known.

2.5.List of Functions Required in the Parent Application

Note: This section applies to full source distribution only.

Function	Description
TxData	Used by the RDS encoder for sending replies to specified port. In the parent application, the data should be placed into a TX buffer and physically sent to the port as soon as possible. The TX buffer free capacity must always be at least twice the *CMD_BUFFER_LEN*. The parent application should not call the *PortRxBuffer* function while the free capacity is lower. Return value (passed to the RDS encoder) indicates whether the TX buffer is still able to accept data (return 0) or there's not enough free capacity for next data (return any non-zero value).

2.6.List of Configurable Parameters in the Sources

Note: This section applies to full source distribution only.

These parameters are located in the file rdsenc.h.

Parameter	Description
TOTAL_PORTS	Specifies how many independent ports the encoder will have. By default, each port allocates about 300 bytes of RAM in the encoder + at least 0.5 kB in the parent application for necessary TX buffer + some space for RX buffer (these buffers can however have dynamically allocated space or may be already present in Ethernet layer etc.).
SYSTEM_PORT	Specifies which of the ports above will be a "system port". System port is not directly visible to the end user but provides a bidirectional communication between the encoder and the parent application, for example sending data from non-volatile memory to the encoder on power-up or setting output RDS level when requested by UECP on another port. There is no acknowledgement sent to the commands on the system port. UECP frames sent via the system port are not CRC-checked if the CRC bytes are set to 0. They are also not checked for length if the MFL byte is set to 0. This simplifies the control from the parent application.
CMD_BUFFER_LEN	Max. length of control command or UECP frame, must be > 260.
MAX_DPS_LEN	Max. length of dynamic PS text.
TOTAL_DSN	Total number of Data sets (DS). Each data set allocates about 3 kB of RAM (depending on other settings). More than 2 DS are used very rarely.
TOTAL_PSN	Total number of Program services (EON) in each DS. By default, each EON allocates about 150 bytes in each DS.

TOTAL_RT	Max. number of Radiotexts in each DS. Radiotexts are transmitted in a loop, as defined by the UECP.
PORT_RX_TIMEOUT	Port RX timeout in seconds. If there's no character received during this period, appropriate port command buffer is cleared.
TOTAL_ADDR_ITEMS	Max. number of items in UECP address list. Two items are usually enough for any purpose.
TOTAL_GS_ITEMS	Max. number of items (groups) in Group sequence.
TOTAL_AF_ITEMS	Max. number of items in AF lists. Large networks using AF method B rarely require more than 256 items.
TOTAL_EONAF_ITEMS	Max. number of items in EON AF lists. Since the time required for complete transmission of all EON information is limited by the RDS standard, the broadcaster usually cannot use more than a few tens of EON AF items, regardless of the network largeness.
TOTAL_FREE_GROUPS	Size of buffers for free-format groups, incl. TDC, IH and EWS. Separate buffer is allocated for each group type.
TOTAL_GROUP_TYPES	Expected maximum number of different group types entered via the free-format feature. Maximum value of 32 is equal to the total number of RDS group types. In real operation, no broadcaster uses more than 10 different group types in total, only some of them are free-format.
TOTAL_TMC_GROUPS	Buffer size for TMC groups.
ENCODERS_DELAY_MS	Tells the RDS encoder the real average delay between calling of the Function for getting output RDS data and occurrence of that data on the output of the transmitter. Thus this constant represents a delay caused by the parent application given by its architecture, output data buffering, modulation process etc. The purpose of the constant is in correction of the internal encoder's RTC clock to keep the time transmitted accurate.
UECP_TIME_VALID_FOR	Arbitrates whether the encoder uses time information entered via UECP or via the *UpdateTime* function as these two times may differ. After receiving a time via UECP, that time is valid for *UECP_TIME_VALID_FOR* minutes and the *UpdateTime* function results in no action during this period.
DEFAULT_PS	Factory default Program Service name (8 characters).
VER_RESPONSE	Response to the VER command, usually it contains fw version.
DEFAULT_GS *DEFAULT_GS_LEN*	Default Group sequence and its length in bytes. May be empty (zero length).

It is recommended to keep all values at their defaults.

2.7. Storing/Restoring the RDS Configuration to/from a Non-volatile Memory

Note: This section applies to full source distribution only.

There's no standard defined for storing/restoring RDS configuration to/from a non-volatile memory (FLASH, EEPROM). This configuration is typically used on power-up. It should contain at least the basic and system parameters, like the station's PS, PI, AF, current DSN (Data set number), port settings etc., but it typically covers also default content of RT, dynamic PS text, etc. Parameters and services which are related only to current broadcast content should not be stored in non-volatile memory (typically TA, TMC, RAW, LI, PIN). The parent application is responsible for all non-volatile memory operations.

The parent application can restore configuration from non-volatile memory using two ways: by direct filling the variables in the RDS Data Structure or by sending configuration commands via the System port.

Similar way, the parent application can store the configuration to non-volatile memory by direct reading the variables from the RDS Data Structure or by asking for the configuration values via the System port. The storing may be initiated by end user via embedded GUI or via ASCII command SAVESET which redirects to the System port.

It is strongly recommended that the parent application works with each variable in the RDS Data Structure separately rather than creating an image of entire structure. Otherwise, possible changes in the structure in future versions may cause the data interpretation failure.

2.8. Writing/Reading the RDS Data Structure

Note: This section applies to full source distribution only.

Accessing the RDS Data Structure directly is useful especially for purposes of showing actual settings, indicate the encoder's status as well as for non-volatile memory operations.

The RDS Data Structure is a set of global single variables and structures which divides into four areas:

- RDS services associated with Data sets
- RDS services independent of Data sets
- Information associated with communication ports
- Internal RDS encoder's settings

This section explains a purpose of the most important variables and structures. For exact format and declaration, follow the C sources.

2.8.1. RDS services associated with Data sets

Variable/*Structure*	**Purpose**
CurrentDSN	Contains index of current Data set. Important variable for accessing RDS services being broadcast. ***Important note:*** *Internally the Data sets are indexed from 0 while the UECP protocol indexes from 1. A Data set with index 1 in the UECP effectively points to Data set with index 0 in the RDS encoder.*
DataSets[]	Contains all data and RDS services associated with Data sets. For example, current Program service name (PS) can be found in DataSets[CurrentDSN].PS

2.8.2. RDS services independent of Data sets

Variable/*Structure*	**Purpose**
FreeGroupBuffer[]	Free-format group buffers.
TMCGroupBuffer	TMC group buffer.
CurrentRT	Index (within the current Data set) of currently transmitted Radiotext.

2.8.3. Information associated with communication ports

Variable/*Structure*	**Purpose**
CommandBuffers[]	Contains independent buffers for incoming data and other operational data associated with each communication port. ***Important note:*** *Internally the Ports are indexed from 0 while the UECP protocol indexes from 1. A Port with index 1 in the UECP effectively points to Port with index 0 in the RDS encoder.*

2.8.4. Internal RDS encoder's settings

Variable/*Structure*	**Purpose**
SiteAddress[]	List of site addresses.
EncoderAddress[]	List of encoder addresses.
SADR_Num, EADR_Num	Number of items in the list of site, respectively encoder addresses.
UTC_MJD, UTC_H, UTC_N, UTC_S, LTO	Internal RDS encoder's clock, local time offset.
MECAccess[256]	Encoder Access Control. For each MEC (index), appropriate integer keeps its enable(1)/disable(0) state. Bit 0 of the integer represents state valid for port 0, bit 1 for port 1 etc.

2.9. Initialization of the Encoder

Note: This section applies to full source distribution only.

After power-up or device reset, the parent application must call the encoder's function *Initialize* prior to calling any other encoder's function or accessing any encoder's data.

Once the encoder is initialized, its content is set to default values:

- **Data sets:** All Data sets (*TOTAL_DSN*) are created with following content: PS set to *DEFAULT_PS*, PI is set to FFFF, Group sequence set to *DEFAULT_GS* (if the Group sequence is empty, it results in groups 0B on the RDS output), all functions related to the main service are disabled, all EON services (*TOTAL_PSN)* are created but kept disabled.
  Active Data set is Data set 1 (internally its index - represented by CurrentDSN - is 0).
- **Communication ports:** Access is enabled on all ports for all UECP commands. Terminal ECHO is disabled on all ports. UECP address list is cleared (excl. fixed values of 0 for encoder address and site address).
- **Others:** Free-format and TMC group buffers are cleared.

After the initialization the parent application may call the *UpdateTime* function and restore the encoder's content by copying data from non-volatile memory (if there is any).

2.10. Code Table Conversion Routines

Note: This section applies to full source distribution only.

Although today's world communicates mostly in Unicode variant UTF-8, the encoder internally works with code table defined in the RDS standard (see Annex 2). The reason is that UECP protocol, as defined, is practically not able to carry text messages in UTF-8 coding. The text fields in the UECP protocols are fixed or limited length while the text coded in UTF-8 has various lengths (in bytes) depending on the text content. All texts that contain national characters should be converted according to the RDS code table before entering to the RDS encoder.

The RDS encoder provides a set of functions which can be used for this purpose in some specific situations. Typically, the user enters a Radiotext via embedded website, the parent application converts the text from Unicode to RDS code page and sends it to the encoder.

These functions are not used internally but are provided for call from the parent application.

Function	**Description**
UnicodeCharToRDS	Converts a character given by 16-bit Unicode representation to the same or nearest character from the RDS code table.
RDSExtendedToBasic	Replaces all extended characters (in range 127-255) in the RDS text to the nearest basic characters (in range 32-127). This function is useful for showing current text messages on a LCD or optionally for ensuring compatibility with all receivers because many receivers are not able to show the extended characters properly.
RDSCharToUnicode	Converts a character from the RDS code table to its Unicode 16-bit equivalent.

The functions are defined in the file rdsconv.c.

3. Configuration Commands

3.1. UECP Commands

For data format, syntax and description, follow the UECP Specification [1].

UECP Command	Purpose
01 PI	To set the PI code of the specified programme service(s) of the specified data set(s).
02 PS	To set the PS name of the specified programme service(s) of the specified data set(s).
04 DI/PTYI	To set the Decoder Information and dymamic PTY Indicator codes of the specified programme service(s) on the specified data set(s).
03 TA/TP	To set the Traffic Announcement and Traffic Programme bits.
05 MS	To set the MS flag of the specified programme service on the specified data set(s).
07 PTY	To set the Programme Type information of the specified programme service on the specified data set(s).
3E PTYN	To set the Programme Type Name of the specified service(s) of the specified data set(s).
0A RT	To edit Radiotext. If the buffer contains only one message, then transmission is repeated indefinitely, despite the defined number of transmissions. If further messages are added, then each message (including the first) is transmitted as defined by its number of transmissions field, within an overall indefinite cycle.
13 AF	To edit AF data in the specified data set(s) of the specified programme service(s).
14 EON-AF	To edit EON AF data on the specified data set(s) of the specified programme service(s).
1A Slow labeling codes	To edit data for type 1A group, block 3.
2E Linkage information	To edit Linkage information, variant 12 of block 3 of type 14A groups. The Linkage activator LA (MSB) is also in type 1A group, block 3.
26 TDC	To edit the data for the Transparent Data Channel.
2B EWS	To edit the Emergency Warning System data (37 bits) in type 9A group.
25 IH	To edit the In-house applications and specify the group version.
30 TMC	To edit the TMC data in type 8A group.
24 Free-format group	To add a group to the free-format buffer for that group type.
0D Real time clock	To set the date and time.
09 Real time clock correction	To set real time clock correction (RTCC) in order to compensate a delay caused by the signal distribution.
19 CT On/Off	To enable/disable the transmission of type 4A group.
1E RDS On/Off	To switch RDS output signal "On" or "Off".

	Note: The request redirected to the System port output as 'CMD:RDS_ON' or 'CMD:RDS_OFF'.
0E RDS level	To adjust the level of the RDS subcarrier in mVpp. *Note: The request redirected to the System port output as 'CMD:RDS_LEVEL_xxxx'.*
23 Site address	To load or remove a site address in the encoder.
27 Encoder address	To load or remove an encoder address in the encoder.
28 Make PSN list	To assign one PSN as the main network service in the specified data set(s) and assign the other PSNs as other networks (EON).
0B PSN enable/disable	To enable or disable a specified PSN.
3F EON elements enable/disable	To enable or disable the transmission of a specific EON element for usage in group type 14 for specific PSNs and DSNs.
1C Data set select	To select desired data set to be active ("on air").
16 Group sequence	To set the group sequence in the specified data set(s).
3A Encoder Access Right	To enable or disable access to any message (command) on any port.
17 Request message	To request specific message to be replied by the encoder.
18 Message acknowledgment	To report error when getting invalid Request message command.

Notes (UECP commands):

1. *RDS level specified in mVpp by the UECP protocol – digital FM modulators not supported by the UECP.*
2. *RDS phase expected to be kept at optimal value (90 deg. compared to pilot) by the DSP unit.*
3. *Non-volatile memory usage: All operations are preferably performed in RAM to prevent the device damage in case of receiving dynamically updated data (typical FLASH endurance is 100k of write cycles).*
   The UECP protocol does not recognize between volatile and non-volatile memory.
4. *If a redirection to the system port is made, the format is 'CMD:<command>'. The command is translated to ASCII representation. Entire sequence is passed to the parent application via function TxData and it is passed at once. The parent application can simply identify it by the 'CMD:' prefix.*

3.2. ASCII Commands

3.2.1. Quick Overview

ASCII Command	Description
TEXT	Updates the Radiotext (If more Radiotexts is present in the RT loop, the update is applied to the first one, that is the RT with index 0).
RT1	
DPS	Updates the Dynamic/scrolling PS (immediately)
DPS1	
DPSENQ	Updates the Dynamic/scrolling PS (using queue)
DPSMOD	Specifies the Dynamic/scrolling PS display mode: 0 – fixed 8 char., 1 – scrolling, 2 – word parsing, 3 – space separated scrolling
DPSEN	Enables/Disables the Dynamic/scrolling PS
LABPER	Dynamic PS speed
SCRLSPD	Scrolling PS speed (high/low)
SPSPER	Static PS period – a delay between two repeats of the Dynamic/scrolling PS
PS	Default static PS (Program Service name).
TPS	Traffic PS (Program Service name during TA and TP = 1).
RTTODPS	If enabled, any new Radiotext is copied into Dynamic/scrolling PS buffer in order to show the Radiotext message also in Dynamic/scrolling PS form. Since there's no UECP command for Dynamic PS, this is the only way how to control the Dynamic PS via the UECP protocol.
SAVESET	Saves general settings or entire content of specified Data set to non-volatile memory. *Note: The request redirected to the System port output as 'CMD:SAVESET'.*
ECHO	Enables/Disables the port echo when typing ASCII commands.
CT	To enable/disable the transmission of type 4A group.
DI	To set the Decoder Information and dymamic PTY Indicator codes of the main programme service on the current data set.
TA	To set the Traffic Announcement bit.
TP	To set the Traffic Programme bit.
MS	To set the MS flag of the main programme service on the current data set.
PTY	To set the Programme Type information of the main programme service on the current data set.
PI	To set the PI code of the main programme service of the current data set.
PROGRAM	To select desired data set to be active ("on air").
VER	To get the encoder's firmware version, incl. possible manufacturer string.
SETSPY	To initiate remote RDS output monitoring.

Notes (ASCII commands):

1. *Command syntax for sending data: COMMAND=VALUE<Enter>, the encoder confirms by CR+LF+'+'+CR+LF+CR+LF*
2. *Command syntax for retrieving data: COMMAND<Enter>, the encoder returns data as CR+LF+VALUE+CR+LF+'+'+CR+LF+CR+LF*
3. *Other more ASCII commands to consider.*
4. *Unlike the UECP protocol, the ASCII protocol for RDS encoders has never been issued as a standard. Details of implementation depend on manufacturer. There are some advantages of ASCII commands – can be entered via a terminal, can control RDS services which are not included in the UECP protocol (dynamic PS, specific services of particular RDS encoder etc.), are sometimes easier to implement in the control application due to plain text form instead of binary data.*

3.2.2. General Syntax

The RDS encoder's ASCII command interpreter meets the following rules:

Any instruction sent to the RDS encoder must be **validated** by <Enter>.
Before validating you may correct the characters by pressing <Backspace>.

Typically there are two methods of use for the commands:

- Setting new value: `Command=Value<Enter>`
- Reading actual value: `Command<Enter>`

Depending on the command processing success, several characters (followed by two pairs of carriage return and line feed characters) can be returned by the RDS encoder:

+	Command processed successfully
!	Unknown command
-	Invalid argument
/	Command processed partially

The command interpreter is not case sensitive.

Example of typical data flow:

RX	**P**	**S**	**=**	**P**	**R**	**O**	**␣**	**8**	**8**	←						
TX (ECHO=1)		**P**	**S**	**=**	**P**	**R**	**O**	**␣**	**8**	**8**	←	↓	**+**	←	↓	←	↓
TX (ECHO=0)											←	↓	**+**	←	↓	←	↓

RX	**P**	**S**	←														
TX (ECHO=1)		**P**	**S**	←	↓	**P**	**R**	**O**	**␣**	**8**	**8**	←	↓	**+**	←	↓	←	↓
TX (ECHO=0)				←	↓	**P**	**R**	**O**	**␣**	**8**	**8**	←	↓	**+**	←	↓	←	↓

Legend:
TX – data sent to the RDS encoder, RX – data read from the RDS encoder, ← - CR (char. 13), ↓ - LF (char. 10)

3.2.3. Description

DPS, DPS1	**Dynamic PS**
A text message to be displayed on receiver instead of static PS name. See the next section for details.	
`DPS=Hello World`	Sets the DPS text
`DPS=`	Clears the DPS

DPSEN	**Dynamic PS Enable**
Enables (1) or Disables (0) the Dynamic/scrolling PS	
`DPSEN=1`	Enables the Dynamic PS feature

DPSENQ	**Dynamic PS Enqueue**
Advanced version of the `DPS` command. Places the text to a one level deep queue. New text will not be displayed on the receiver until old text reaches its end.	
`DPSENQ=Hello World`	Sets the following DPS text

DPSMOD	**Dynamic PS Mode**	**(0-3)**
Display mode for the DPS1 text. 0 - Scrolling by 8 characters 1 - Scrolling by 1 character 2 - Word parsing mode 3 - Scrolling by 1 character, text separated by spaces at begin and end		
`DPS1MOD=3`		

RTTODPS	**Copy Radiotext to Dynamic PS**	**(0, 1)**
If enabled (1), any new Radiotext is also copied into Dynamic/scrolling PS buffer in order to show the Radiotext message also in Dynamic/scrolling PS form.		

`RTTODPS=1`
`TEXT=Hello World`
`DPS`		

LABPER	**Label Period**	**(0-255)**
Label Period used in DPS Mode 0 and 2. Increasing the value by 1 increases the period by approx. 0.6 seconds (exact value depends on Group Sequence).		
`LABPER=4`	Each label is displayed for about 2 seconds.	

PS	**Program Service name**
Static name of radio station that is displayed on receiver. Max. 8 characters long.	
`PS=KISS FM`	

TPS	**Traffic PS**
Static text displayed on receiver during traffic announcements (TA and TP = 1)	
`TPS=Traffic`	
`TPS=`	Disables the Traffic PS

TEXT, RT1	**Radiotext**
Up to 64 characters long text message to be displayed on receiver in Radiotext format. Primarily used for song titles streaming etc. Car radios usually don't support this service, Dynamic PS can be used instead.	
`TEXT=Hello World`	

SCRLSPD	**Scrolling PS Speed**	**(0, 1)**
Sets high (1) or low (0) speed of scrolling PS transmission. Although setting high speed gives the result looking better, remember that on some receivers or under bad reception conditions the text may be unreadable. The reason is absolutely outside the RDS encoder and comes out from the fact that scrolling PS has never been included in RDS standard. Due to this the high speed is not recommended.		
`SCRLSPD=1`		

SPSPER	**Static PS Period**	**(0-255)**
Specifies the time between two repeats of the Dynamic PS text. Default static PS (station name) is displayed during this time. Increasing the value by 1 increases the period by approx. 0.6 seconds (exact value depends on Group sequence).		
`SPSPER=10`	Sets the period duration to about 6 seconds.	

SAVESET	**Save to EEPROM**
Store Content of the Current Data Set to a non-volatile EEPROM/FLASH memory. *Internally, this command is only redirected to the output of the System port. The parent application, after receiving this command, is responsible for making the operation.*	
`SAVESET`	

ECHO	**Terminal Echo**	**(0, 1)**
Determines if the RDS encoder sends an echo (1) of each character or not (0) on appropriate port. *This feature is intended for manual control of the encoder via terminal application. The encoder does not send echo characters when communicating with a machine, i.e. when processing large blocks of incoming data at once.*		
`ECHO=1`		

DI	**Decoder Identification**	**(0-15)**
Identification of the decoder to be used by the receiver.		
`DI=1`	Standard transmission – stereo.	
`DI=0`	Standard transmission – mono.	

MS	**Music/Speech**	**(0, 1)**
Music/Speech switch.		
`MS=0`	Speech program	
`MS=1`	Music program	

PI	**Program Identification**	**H (1000-FFFF)**
Identification code of the radio station. Always contains four hexadecimal digits.		
`PI=20FE`	OK	
`PI=0F55`	Not allowed in real operation (0 as first digit)	

PTY	**Program Type number**	**(0-31)**

An identification number to be transmitted with each program item, intended to specify the current Program Type within 31 possibilities.

Program type codes (Europe):

0 - (none)	16 - Weather
1 - News	17 - Finance
2 - Affairs	18 - Children
3 - Info	19 - Social Affairs
4 - Sport	20 - Religion
5 - Education	21 - Phone In
6 - Drama	22 - Travel
7 - Cultures	23 - Leisure
8 - Science	24 - Jazz Music
9 - Varied Speech	25 - Country Music
10 - Pop Music	26 - National Music
11 - Rock Music	27 - Oldies Music
12 - Easy Music	28 - Folk Music
13 - Light Classics Music	29 - Documentary
14 - Serious Classics	30 - Alarm Test
15 - Other Music	31 - Alarm

Program type codes (US RBDS):

0 - (none)	16 - Rhythm and Blues
1 - News	17 - Soft Rhythm and Blues
2 - Information	18 - Foreign Language
3 - Sports	19 - Religious Music
4 - Talk	20 - Religious Talk
5 - Rock	21 - Personality
6 - Classic Rock	22 - Public
7 - Adult Hits	23 - Leisure
8 - Soft Rock	24 - College
9 - Top 40	25 - (unassigned)
10 - Country	26 - (unassigned)
11 - Oldies	27 - (unassigned)
12 - Soft	28 - (unassigned)
13 - Nostalgia	29 - Weather
14 - Jazz	30 - Emergency Test
15 - Classical	31 - Emergency

`PTY=10` Sets the Pop Music Program Type (EU)

TA	**Traffic Announcement**	**(0, 1)**

Indicates instantaneous presence (1) of traffic information during broadcasting.

`TA=1`

TP	**Traffic Program**	**(0, 1)**

This is a flag to indicate that the tuned program carries traffic announcements. The TP flag must only be set on programs that dynamically switch on the TA identification during traffic announcements. The signal shall be taken into account during automatic search tuning.

`TP=1`

CT	**Clock Time and Date**	**(0, 1)**

Enables (1) or disables (0) time and date transmission in CT format.

```
CT=1
```

PROGRAM	**Data Set Selection**	**(1-TOTAL_DSN)**

To select desired data set to be active ("on air").

```
PROGRAM=1
```

SETSPY	**Set RDS Monitoring Counter**	**(1-255)**

Specifies how many output RDS groups will be copied to the port from which the command is called.
For continuous monitoring, the command must be send again before the counter elapses.

RDS data is provided by the RDS encoder in ASCII representation in this format:

```
"G:"+#13+#10+"AAAABBBBCCCCDDDD"+#13+#10+#13+#10
```

where

AAAA is PI,
BBBB is block 2,
CCCC is block 3,
DDDD is block 4 of the RDS group.

No CRC or offsets are included.

`SETSPY=114`	Send 114 RDS groups (that is, monitor the RDS output for 10 seconds)

3.2.4. Dynamic/Scrolling PS in Details

Standard RDS enabled receiver disposes of 8-character LCD display but we sometimes need to show pile of information and commercials. So small display on the one hand and so much demands on the other hand. Although Radiotext service is defined in the RDS standard, this service is not present some receivers (especially older car radios) and has some other limitations. According to the broadcasters needs, the PS service - one of the basic RDS services supported by all receivers - can be usually used to give sequential information. This has become known as 'Dynamic PS' or 'Scrolling PS'.

Note: Using the dynamic PS is restricted in some countries and it's fully prohibited by the RDS standard!

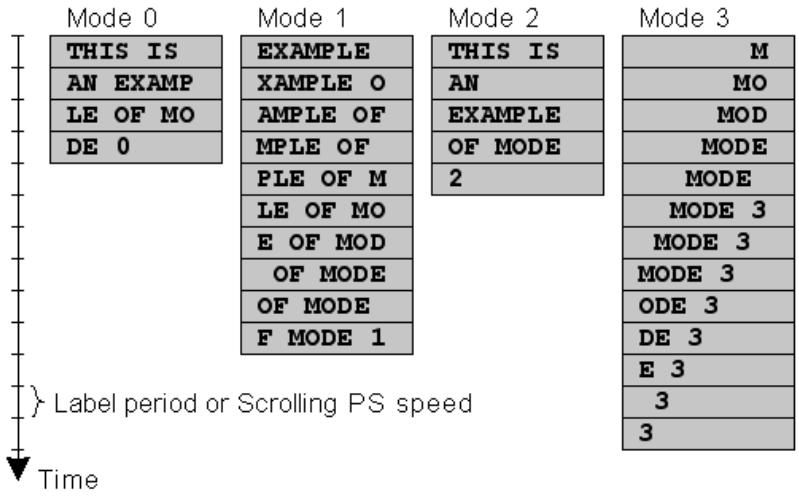
Note: Nowadays, the dynamic PS gradually loses its meaning as most of the currently produced receivers support Radiotext.

Note: The manufacturer is not responsible for incompetent use of the dynamic PS feature. Some receivers may not display the dynamic PS properly for reasons that lie entirely on their side. Never provide traffic information inside the Dynamic PS text!

Basic dynamic PS text message length is up to 128 characters (mode independent). Basic configurable parameters are:
- Text content
- Display mode
- Label period or scrolling speed

Static PS period (delay between text loops) specifies the time between two repeats of the Dynamic PS text loops. Default PS is displayed during this time.

Four display modes are provided. The mode is switchable 'on the fly', without need to re-enter the text message.
- Mode 0 - Scrolling by 8 characters
- Mode 1 - Scrolling by 1 character
- Mode 2 - Word parsing scrolling
- Mode 3 - Scrolling by 1 character, text separated by spaces at begin and end



4. Annexes

4.1. RDS Data Structure

4.1.1. RDS Group Types and Assignment

Group type	Binary coding	HEX coding *	RDS services
0A	00000	00	**TA, MS, DI, AF, PS**
0B	00001	01	**TA, MS, DI, PS**
1A	00010	02	**ECC, LIC, PIN**
1B	00011	03	**PIN**
2A	00100	04	**RT** (64 characters)
2B	00101	05	**RT** (32 characters)
3A	00110	06	**AID for ODA**
3B	00111	07	ODA
4A	01000	08	**CT**
4B	01001	09	ODA
5A	01010	0A	**TDC** or ODA
5B	01011	0B	**TDC** or ODA
6A	01100	0C	**IH** or ODA
6B	01101	0D	**IH** or ODA
7A	01110	0E	**RP** or ODA
7B	01111	0F	ODA
8A	10000	10	**TMC** or ODA
8B	10001	11	ODA
9A	10010	12	**EWS** or ODA
9B	10011	13	ODA
10A	10100	14	**PTYN**
10B	10101	15	ODA
11A	10110	16	ODA
11B	10111	17	ODA
12A	11000	18	ODA
12B	11001	19	ODA
13A	11010	1A	**ERP** or ODA
13B	11011	1B	ODA
14A	11100	1C	**EON**
14B	11101	1D	**EON TA**
15A	11110	1E	(not defined)
15B	11111	1F	**TA, MS, DI**

* Note: Format required by UECP command 16 (Group sequence)

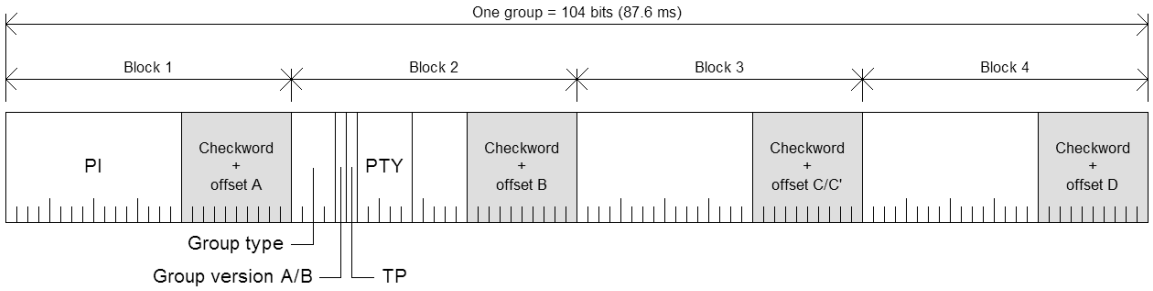
4.1.2. RDS Group Format

Following information is provided for better understanding to the RDS principles and the user defined group coding.

The largest element in the RDS coding structure is called a "group" consisting of 104 bits. The group comprises 4 blocks of 26 bits each. Each block comprises an information word and a checkword. Each information word comprises 16 bits. Each checkword comprises 10 bits.

All information words, checkwords, binary numbers or binary address values have their most significant bit (MSB) transmitted first.

The data transmission is fully synchronous and there are no gaps between the groups or blocks. The basic data-rate of the system is 1187.5 bit/s. Thus transmission of one group takes about 87.6 ms and about 11.4 groups are transmitted per one second.



General RDS group format.

Basic principles and rules

- The services which are to be repeated most frequently, and for which a short acquisition time is required (PI, TP, PTY), in general occupy the same fixed positions within every group.
- **There is no fixed rhythm of repetition of the various types of group**, i.e. there is ample flexibility to interleave the various kinds of message to suit the needs of the users at any given time.
- The first four bits of the second block of every group are allocated to a four-bit code which specifies the application of the group - group type. Groups are referred to as types 0 to 15.
- For each type (0 to 15) two "versions" can be defined. The "version" is specified by the fifth bit of block 2: 0 = version A, 1 = version B.
- For all groups of version B the PI is inserted also in block 3 so this block cannot carry any other information when version B of the group is used.

4.2.RDS Character Set

The RDS system does not support Unicode character set. Standard character set (G0) used within the RDS system is limited to the characters provided in following EBU Latin code-table.

0x	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0	@	P	`	p	á	â	ª	º	Á	Â	Ã	ã
1	!	1	A	Q	a	q	à	ä	α	¹	À	Ä	Å	å
2	"	2	B	R	b	r	é	ê	©	²	É	Ê	Æ	æ
3	#	3	C	S	c	s	è	ë	‰	³	È	Ë	Œ	œ
4	¤	4	D	T	d	t	í	î	Ğ	±	Í	Î	ŷ	ŵ
5	%	5	E	U	e	u	ì	ï	ě	İ	Ì	Ï	Ý	ý
6	&	6	F	V	f	v	ó	ô	ň	ń	Ó	Ô	Õ	õ
7	'	7	G	W	g	w	ò	ö	ő	ű	Ò	Ö	Ø	ø
8	(	8	H	X	h	x	ú	û	π	µ	Ú	Û	Þ	þ
9	)	9	I	Y	i	y	ù	ü	₤/€	¿	Ù	Ü	Ŋ	ŋ
A	*	:	J	Z	j	z	Ñ	ñ	£	÷	Ř	ř	Ŕ	ŕ
B	+	;	K	[	k	{	Ç	ç	$	°	Č	č	Ć	ć
C	,	<	L	\	l	|	Ş	ş	←	¼	Š	š	Ś	ś
D	-	=	M	]	m	}	β/ß	ğ	↑	½	Ž	ž	Ź	ź
E	.	>	N	^	n	~	¡	ı	→	¾	Đ	đ	Ŧ	ŧ
F	/	?	O	_	o		IJ	ij	↓	§	Ŀ	ŀ	ð	

Basic set

Extended set

Example of use:

Character 'é' can be found in the table on coordinates 8, 2. Thus it converts to ASCII character 0x82 (HEX) or 130 (decimal) that must be sent to the RDS encoder.

Notes:

- *Many commercially available receivers use 14-segment LCD displays. These receivers are able to display only a limited part of the basic character set (green framed) and do not support the extended set at all. All lower-case characters are showed as upper-case.*
- *The EBU Latin basic set is compatible with most of other systems so conversion is not required for this region.*
- *Since many receivers contain no support for the extended character set, it is recommended to keep all characters in all text messages in the basic set region.*
- *Auxiliary code-tables (G1, G2) are not discussed in this document as they are not intended for common use due to lack of support on most receivers and incorrect interpretation of repertoire control characters on such receivers.*

4.3. Demo Encoder

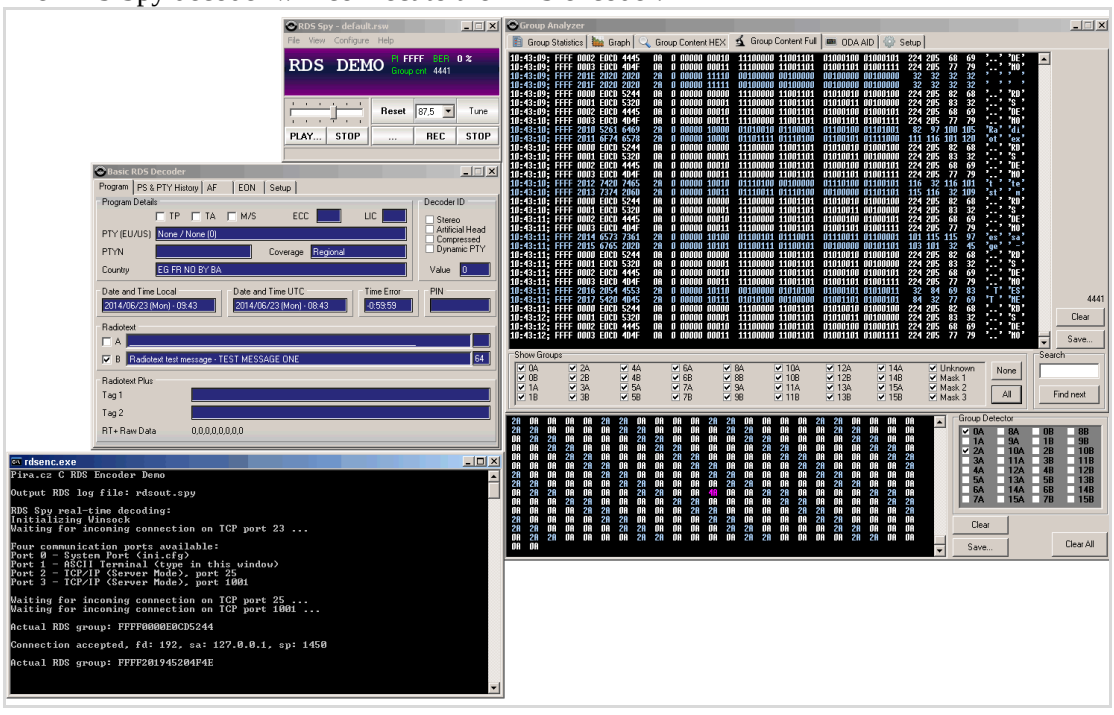
The demo encoder represents an example of use of the RDS encoder functions in Windows. The demo encoder is included in the Magic RDS 4 installation package.

The RDS output is viewable in real time via the RDS Spy decoder. Several methods of RDS encoder control are provided.

4.3.1. Getting it working

1. Download and install the **Magic RDS 4** (http://pira.cz/rds/).
2. Run the application Magic RDS 4 and double-click on **'Add new connection'** icon.
3. In the dialogue box, select **RDS Encoder**, device model: **Demo encoder**. Confirm by **Add** button.
4. Click on **'Run Demo Encoder'** button.
5. Click on the **RDS Spy** icon:

The RDS Spy decoder will connect to the RDS encoder:



6. Configure the RDS encoder from the application or create a Virtual port and connect a data source (terminal, virtual serial port, broadcast automation system etc.) thru it.

4.3.2. Other features of the demo application

Note: This section applies to full source distribution only.

Command line options:

`rdsdemo.exe file1` - processes content of the file1 on start-up via system port

ASCII terminal:

Keyboard input and screen output is connected to control port 1 of the encoder. That means a simple terminal is available, allowing ASCII communication with the encoder via its port 1 (for example, type PS=KISS FM<CR> or PS<CR>).

TCP ports:

Control ports 2 and 3 are associated with simple TCP servers based on Windows Sockets. Any TCP client can connect to these ports.

Output data logging:

Output RDS data are saved to a *.spy file, it may be played later using the RDS Spy decoder.

4.3.3. Simple UECP testing application (uecptest.exe)

This application allows checking the RDS encoder's behavior and responses depending on the UECP commands entered. You'll need the [1] for proper working with this application.

First steps:

1. Run the RDS encoder (either the exe file or the real product).
2. Fill the connection parameters and click Connect.
3. If necessary, configure the address items, DSN and PSN, leave 0 otherwise.
4. In the Message element editor, select the command you want to test. The command structure appears. The user specifies only the MED fields, other fields are calculated and entered automatically.
5. Click on 'Send directly' if you want to test only this command. Complete UECP message will be created and sent to the encoder.
6. Click on 'Add to MSG' if you want to create a UECP frame with multiple Message elements. When done, click on 'Send'.
7. The Log window shows the data sent and received. (Responses are received upon UECP command '17: Request message'.)

Examples:

To set PI for main service of Data set 1: Set DSN to 1, PSN to 0, select MEC '01: PI', enter the PI in the MED field, e.g. **`20 FA`**.

To get the PI if main service of current Data set: Select MEC '17: Request message', to the **MED field**, enter the PI MEC 01, DSN 01 and PSN 00: **`01 01 00`**