Technical Information Manual

Revision n. 2
17 January 2007

CAENRFIDLib

ANSI C FUNCTIONS LIBRARY

NPO:
00117/03:RFLIB.MUTx/02

INDEX

1. Introduction

The CAEN Long Range UHF Readers are developed in Europe and in compliance with European and US telecommunication regulations, are a step forward in UHF RFID readers. Capable of long distance reading using extremely low RF energy, the CAEN Long Range UHF Readers are optimized to increase receiver sensibility and reduce transmitter noise.
The CAEN Long Range UHF Readers' open architecture uses a multi-protocol technology. The tag protocol interface was developed using a field programmable gate array, which allows easy modification of the tag protocol definition. On the host side, a powerful 32bit micro-controller enables fast firmware updating for maximum upgradeability to future generations of the EPC specification. Easily integrated with most popular database software and fully compliant with ISO 18000-6B and EPC Class 1 – Generation 1 protocol, Philips UCODE EPC 1.19, the CAEN Long Range UHF Readers can be used in conjunction with any passive or active tag that conforms to the same standards; moreover, it can be easily upgraded for compliancy with other protocols[1].
With their extended read range, the CAEN Long Range UHF Readers are well suited to asset management and logistics applications that require the simultaneous reading of a large number of tags from a great distance.

1.1. CAENRFIDLib introduction

This section describes the CAENRFIDLib library and its implemented functions. CAENRFIDLib is a set of ANSI C functions which permits an user program the use and the configuration of the CAEN Long Range UHF Readers.
The present description refers to CAENRFIDLib, available in the following formats:

– Win32 DLL (CAEN provides the CAENRFIDLib.lib stub for Microsoft Visual C++ 6.0)

CAENRFIDLib is logically located between an application like the samples provided and the lower layer software libraries.

1.2. CAENRFIDLib description

1.2.1. CAENRFIDTypes.h

```
#define MAX_ID_LENGTH 12
typedef int CAENRFIDHandle;

/*
    Error codes
*/
typedef enum {
    CAENRFID_StatusOK        =  0,  // Operation completed successfully
    CAENRFID_PortError       = -1,  // Error on selected port
    CAENRFID_ParityError     = -2,  // Parity error
    CAENRFID_InitError       = -3,  // Error on init
    CAENRFID_StatusByteError  = -4,  // Error on status byte
```

[1] Software upgrades will be available at: http://www.caen.it/rfid/

```
    CAENRFID_InvalidParam      = -5,  // Invalid parameter error
    CAENRFID_TimeOutError      = -6,  // Time out error
    CAENRFID_Max4Byte          = -7,  // Data length greater than 4
    CAENRFID_PowerOutRange     = -8,  // Power out of range
    CAENRFID_BadAntenna                = -9,  // Antenna not connected
    CAENRFID_GenericError          = -10, // Generic error
    CAENRFID_InvalidHandle         = -11  // Invelid Handle
} CAENRFIDErrorCodes;

/*
      ID length enum
*/
typedef enum {
    L64bit   = 8,
    L96bit  = 12
} CAENRFIDLenghtID;

/*
      Communication ports enum
*/
typedef enum {
    RS232   = 0,
    RS485   = 1,
    TCP     = 2,
    USB     = 3
} CAENRFIDPort;

/*
      Antenna select enum
*/
typedef enum {
    NOANT   = 0,
    ANT1    = 1,
    ANT2    = 2,
    ANT3    = 3,
    ANT4    = 4
} CAENRFIDAntenna;

/*
      Tag identification struct: for each tag it contains
      the ID, the length of the ID and the antenna used to
      identify the tag.
*/
typedef struct {
   byte             ID[MAX_ID_LENGTH];
   int              Length;
   CAENRFIDAntenna     Antenna;
} CAENRFIDTag;

/*
      General purpose outputs masks
*/
typedef enum {
    GPO0    = 0x01,
```

```
    GPO1    = 0x02,
    GPO2    = 0x04,
    GPO3    = 0x08,
} CAENRFIDGpo;

/*
      Bit rate modulation control enum
*/
typedef enum {
    TX10RX10       = 0,
    TX10RX40       = 1,
    TX40RX40       = 2,
    TX40RX160      = 3
} CAENRFIDTxRxConf;


#ifndef CAENRFID_ODL

/*
      RF field control enum
*/
typedef enum {
    CARRIER_OFF    = 0,
    CARRIER_ON     = 1
} CAENRFIDControl;

/*

*/
typedef enum {
    STANDBY = 0,
    ACTIVE  = 1
} CAENRFIDPas;

/*
      Command mode control enum
*/
typedef enum {
    SINGLE         = 0,
    START_SEQ      = 1,
    END_SEQ        = 2,
    SUSTAINED      = 3
} CAENRFIDSetCMD;

#endif
```

1.3. CAENRFIDLib Functions description

1.3.1. CAENRFID_Init

Name: CAENRFID_Init

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function generates an opaque handle to identify a module attached to the PC.

Parameters: [in] Port: Communication port (see CAENRFIDPort enum).
[in] Address: Communication address (i.e.: "COM1" for RS232, "USB0" for USB of IP address for TCP/IP etc.)
[out] Handle: The handle that identifies the device.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Init (CAENRFIDPort Port, char *Address, CAENRFIDHandle *Handle, CAENRFIDProtocol *Protocol);

1.3.2. CAENRFID_ End

Name: CAENRFID_End

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: Notifies the library the end of work and free the allocated resources

Parameters: [in] Handle: The handle that identifies the device

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_End(CAENRFIDHandle Handle);

1.3.3. CAENRFID_GetSWRelease

Name: CAENRFID_GetSWRelease

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: Permits to read the software release of the library.

Parameters: [out] SwRel: Returns the software release of the library

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetSWRelease(char *SwRel);

1.3.4. CAENRFID_GetFWRelease

Name: CAENRFID_GetFWRelease

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: Permits to read the firmware release loaded into the device

Parameters: [in] Handle: The handle that identifies the device.
[out] FWRel: Returns the firmware release of the device

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetFWRelease(CAENRFIDHandle Handle, char *FWRel);

1.3.5. CAENRFID_Inventory

Name: CAENRFID_Inventory

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function returns all the IDs of the tags under the reader fieldusing all the available antennae. The Tags array contains The IDs together with other information related to the single ID such as the antenna under which is the ID and the format of the ID itself (see CAENRFIDTag struct for the details).

Parameters: [in] Handle: The handle that identifies the device.
[in] LogicalSourceName: The name that identify the Logical Source
[out] Tags: Returns an array containing the tags read.
[out] TagsNo: Returns the number of tags in the array.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Inventory(CAENRFIDHandle Handle, char *LogicalSourceName, CAENRFIDTag **Tags, int *TagsNo);

1.3.6. CAENRFID_SetPower

Name: CAENRFID_Inventory

Reader: A928EU, A948EU

Description: The function permits to set the RF field power relative to the antenna socket

Parameters: [in] Handle: The handle that identifies the device.
[in] Power: RF field power expressed in mW.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetPower(CAENRFIDHandle Handle, unsigned int Power);

1.3.7. CAENRFID_Read

Name: CAENRFID_Read

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: This function allows to read Length bytes from the memory of a specific tag identified by the ID (regardless of its status) at the address specified by Address.

Parameters:
[in] Handle: The handle that identifies the device.
[in] ID: The tag ID.
[in] Address: The address of the memory to read.
[in] Length: The number of bytes to read.
[out] Data: The data read from the tag's memory.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Read(CAENRFIDHandle Handle, CAENRFIDTag *ID, int Address, int Length, void *Data);

1.3.8. CAENRFID_Write

Name: CAENRFID_Write

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: This function allows to write Length bytes to the memory of a specific tag identified by the ID (regardless of its status) at the address specified by Address.

Parameters:
[in] Handle: The handle that identifies the device.
[in] ID: The tag ID.
[in] Address: The address of the memory to write.
[in] Length: The number of bytes to write.
[in] Data: The data to write in the tag's memory.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Write(CAENRFIDHandle Handle, CAENRFIDTag *ID, int Address, int Length, void *Data);

1.3.9. CAENRFID_Lock

Name: CAENRFID_Lock

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: This function allows to lockthe memory of a specific tag identified by the ID at the address specified by Address.

Parameters:
[in] Handle: The handle that identifies the device.
[in] ID: The tag ID.
[in] Address: The address of the memory to write.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Lock(CAENRFIDHandle Handle, CAENRFIDTag *ID, int Address);

1.3.10. CAENRFID_TestMode

Name: CAENRFID_TestMode

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to enable/disable te TestMode.

Parameters: [in] Handle: The handle that identifies the device.
[in] TestMode: 0 Disable TestMode >0 Enable

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_TestMode(CAENRFIDHandle handle, unsigned int TestMode);

1.3.11. CAENRFID_SetModulation

Name: CAENRFID_SetModulation

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to control to choose the modulation (the bit rate of the transmission and receive)

Parameters: [in] Handle: The handle that identifies the device.
[in] TxRxCfg: Modulation setting.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetModulation(CAENRFIDHandle Handle, unsigned short TxRxCfg);

1.3.12. CAENRFID_GetModulation

Name: CAENRFID_GetModulation

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to retrieve the modulation (the bit rate of the transmission and receive).

Parameters: [in] Handle: The handle that identifies the device.
[out] TxRxCfg: Modulation setting

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetModulation(CAENRFIDHandle handle, unsigned short *TxRx);

1.3.13. CAENRFID_AllocateChannel

Name: CAENRFID_AllocateChannel

Reader: A928EU, A948EU

Description: The function permits to allocate a notification Channel

Parameters: [in] Handle: The handle that identifies the device.
[in] ChannelName: The Name of the Channel.
[in] ChannelAddress: The Address of the Channel in the form [TCP|USB|RS232]://[ip address:port]

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_AllocateChannel(CAENRFIDHandle handle, char *ChannelName, char *ChannelAddress);

1.3.14. CAENRFID_DeallocateChannel

Name: CAENRFID_DeallocateChannel

Reader: A928EU, A948EU

Description: The function permits to Deallocate a Channel.

Parameters: [in] Handle: The handle that identifies the device.
[in] ChannelName: The Name of the Channel.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_DeallocateChannel(CAENRFIDHandle handle, char *ChannelName);

1.3.15. CAENRFID_AddSourceToChannel

Name: CAENRFID_AddSourceToChannel

Reader: A928EU, A948EU

Description: The function permits to add a LogicalSource to a notification Channel.

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The Name of the Logical Source.
[in] ChannelName: The Address of the Channel.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_AddSourceToChannel(CAENRFIDHandle handle, char *SourceName, char *ChannelName);

1.3.16. CAENRFID_RemoveSourceFromChannel

Name: CAENRFID_RemoveSourceFromChannel

Reader: A928EU, A948EU

Description: The function permits to remove a LogicalSource from a

notification Channel

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The Name of the Logical Source.
[in] ChannelName: The Address of the Channel.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_RemoveSourceFromChannel(CAENRFIDHandle handle, char *SourceName, char *ChannelName);

1.3.17. CAENRFID_AddReadPoint

Name: CAENRFID_AddReadPoint

Reader: A928EU, A948EU

Description: The function permits to add a read point (antenna) to a logical source

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] ReadPoint: The name of the Read Point.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_AddReadPoint(CAENRFIDHandle handle, char *SourceName, char *ReadPoint);

1.3.18. CAENRFID_RemoveReadPoint

Name: CAENRFID_RemoveReadPoint

Reader: A928EU, A948EU

Description: The function permits to remove a read point (antenna) frpm a logical source

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] ReadPoint: The name of the Read Point.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_RemoveReadPoint(CAENRFIDHandle handle, char *SourceName, char *ReadPoint);

1.3.19. CAENRFID_AllocateTrigger

Name: CAENRFID_AllocateTrigger

Reader: A928EU, A948EU

Description: The function permits to create a trigger of the specified type

Parameters: [in] Handle: The handle that identifies the device.
[in] TriggerName: The name of the trigger.
[in] TriggerType: The type of the trigger.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall

CAENRFID_AllocateTrigger(CAENRFIDHandle handle, char *TriggerName, char *TriggerType);

1.3.20. CAENRFID_DeallocateTrigger

Name: CAENRFID_DeallocateTrigger

Reader: A928EU, A948EU

Description: The function permits to destroy a trigger

Parameters: [in] Handle: The handle that identifies the device.
[in] TriggerName: The name of the trigger

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_DeallocateTrigger(CAENRFIDHandle handle, char *TriggerName);

1.3.21. CAENRFID_AddReadTrigger

Name: CAENRFID_AddReadTrigger

Reader: A928EU, A948EU

Description: The function permits to associate a trigger to a source in order to start a read cycle

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] TriggerName: The name of the trigger

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_AllocateTrigger(CAENRFIDHandle handle, char *TriggerName, char *TriggerType);

1.3.22. CAENRFID_RemoveReadTrigger

Name: CAENRFID_RemoveReadTrigger

Reader: A928EU, A948EU

Description: The function permits to remove the read trigger from the logical source

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] TriggerName: The name of the trigger

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_RemoveReadTrigger(CAENRFIDHandle handle, char *SourceName, char *TriggerName);

1.3.23. CAENRFID_AddNotifyTrigger

Name: CAENRFID_AddNotifyTrigger

Reader: A928EU, A948EU

Description: The function permits to associate a trigger to a channel in order

to start a notification.

Parameters: [in] Handle: The handle that identifies the device.
[in] ChannelName: The Address of the Channel.
[in] TriggerName: The name of the trigger.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_AddNotifyTrigger(CAENRFIDHandle handle, char *ChannelName, char *TriggerName);

1.3.24. CAENRFID_RemoveNotifyTrigger

Name: CAENRFID_RemoveNotifyTrigger

Reader: A928EU, A948EU

Description: The function permits to remove the notification trigger from a channel.

Parameters: [in] Handle: The handle that identifies the device.
[in] ChannelName: The Address of the Channel.
[in] TriggerName: The name of the trigger.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_RemoveNotifyTrigger(CAENRFIDHandle handle, char *ChannelName, char *TriggerName);

1.3.25. CAENRFID_GetNotification

Name: CAENRFID_RemoveNotifyTrigger

Reader: A928EU, A948EU

Description: The function permits to decode data coming from the notification channel

Parameters: [in] Skt: The handle to the TCP socket.
[out] Items: A list of data items.
[out] NoItems: The number of data items in the list.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetNotification(SOCKET Skt, CAENRFIDNotify **Items, int *NumberItems);

1.3.26. CAENRFID_GetPower

Name: CAENRFID_GetPower

Reader: A928EU, A948EU

Description: The function returns the value of the ERP power setting in the reader

Parameters: [in] Handle: The handle that identifies the device.
[out] Power: The ERP power of the reader.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID GetPower(CAENRFIDHandle Handle, unsigned int

*Power);

1.3.27. CAENRFID_SetProtocol

Name: CAENRFID_SetProtocol.

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to change the tag protocol used by the reader

Parameters: [in] Handle: The handle that identifies the device.
[in] Protocol: The tag protocol to be set in the reader.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetProtocol (CAENRFIDHandle Handle, CAENRFIDProtocol Protocol);

1.3.28. CAENRFID_GetProtocol

Name: CAENRFID_GetProtocol

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to know what tag protocol is used by the reader

Parameters: [in] Handle: The handle that identifies the device.
[out] Protocol: The tag protocol to be set in the reader.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetProtocol(CAENRFIDHandle Handle, int *Protocol);

1.3.29. CAENRFID_GetReadPointStatus

Name: CAENRFID_GetReadPointStatus

Reader: A928EU, A948EU

Description: The function permits to check the status of a read point

Parameters: [in] Handle: The handle that identifies the device.
[in] ReadPoint: The name of the Read Point.
[out] Status: The status of the read point.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetReadPointStatus(CAENRFIDHandle Handle, char *ReadPoint, CAENRFIDReadPointStatus *Status);

1.3.30. CAENRFID_GetSourceInChannel

Name: CAENRFID_GetSourceInChannel

Reader: A928EU, A948EU

Description: The function permits to check if a logical source is associated to

a specified notification channel that is, the data read from the source is sent to the channel.

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] ChannelName: The name of the Channel.
[out] isPresent: A flag indicating if the source is associated to the specified channel.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetSourceInChannel(CAENRFIDHandle Handle, char *SourceName, char *ChannelName, short *isPresent);

1.3.31. CAENRFID_GetSourceInTrigger

Name: CAENRFID_GetSourceInTrigger

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to check if a logical source is associated to a specified trigger.

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] TriggerName: The name of the Trigger.
[out] isPresent: A flag indicating if the source is associated to the specified trigger.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetSourceInTrigger(CAENRFIDHandle Handle, char *SourceName, char *TriggerName, short *isPresent);

1.3.32. CAENRFID_GetTriggerInChannel

Name: CAENRFID_GetTriggerInChannel

Reader: A928EU, A948EU

Description: The function permits to check if a trigger is associated to a specified notification channel.

Parameters: [in] Handle: The handle that identifies the device.
[in] TriggerName: The name of the Trigger.
[in] ChannelName: The name of the ChannelName.
[out] isPresent: A flag indicating if the trigger is associated to the specified channel.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetTriggerInChannel(CAENRFIDHandle Handle, char *TriggerName, char *ChannelName, short *isPresent);

1.3.33. CAENRFID_GetChannelInTrigger

Name: CAENRFID_GetChannelInTrigger

Reader: A928EU, A948EU

Description: The function permits to check if a channel is associated to a specified notification trigger.

Parameters: [in] Handle: The handle that identifies the device.
[in] ChannelName: The name of the ChannelName.
[in] TriggerName: The name of the Trigger.
[out] isPresent: A flag indicating if the channel is associated to the specified trigger.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetChannelInTrigger(CAENRFIDHandle Handle, char *ChannelName, char *TriggerName, short *isPresent);

1.3.34. CAENRFID_GetReadPointInSource

Name: CAENRFID_GetReadPointInSource

Reader: A928EU, A948EU

Description: The function permits to check if a read point is associated to a specified logical source that is, the read point is used within a read cycle performed in the source.

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The name of the Logical Source.
[in] ReadPoint: The name of the Read Point.
[out] isPresent: A flag indicating if the read point is associated to the specified source.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetReadPointInSource(CAENRFIDHandle Handle, char *ReadPoint, char *SourceName, short *isPresent);

1.3.35. CAENRFID_SetNetwork

Name: CAENRFID_SetNetwork

Reader: A928EU, A948EU

Description: The function permits to configure the network address, the netmask and the default gateway of the reader. The settings are activated after a reboot of the reader.

Parameters: [in] Handle: The handle that identifies the device.
[in] IPAddress: The IP address to set in the form XXX.XXX.XXX.XXX
[in] NetMask: The netmask to set in the form XXX.XXX.XXX.XXX
[in] Gateway: The Gateway to set in the form XXX.XXX.XXX.XXX

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetNetwork(CAENRFIDHandle Handle, char *IPAddress, char *NetMask, char *Gateway);

1.3.36. CAENRFID_SetDE_SB

Name: CAENRFID_SetDE_SB

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to enable the use of the data exchange status bit in the ISO18000-6b anticollision algorithm.

Parameters: [in] Handle: The handle that identifies the device.
[in] Enable: Enable flag.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetDE_SB(CAENRFIDHandle Handle, unsigned int Enable);

1.3.37. CAENRFID_GetDE_SB

Name: CAENRFID_GetDE_SB

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to know if the data exchange status bit is used in the ISO18000-6b anticollision algorithm.

Parameters: [in] Handle: The handle that identifies the device.
[out] Status: The status flag.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetDE_SB(CAENRFIDHandle handle, unsigned short *Status);

1.3.38. CAENRFID_ProgramID

Name: CAENRFID_ProgramID

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to program an EPC Class 1 Gen 1 tag

Parameters: [in] Handle: The handle that identifies the device.
[in] TagID: The EPC to program in the tag.
[in] Password: The kill password to program in the tag.
[in] Lock: Aflag indicating if the EPC has to be locked.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_ProgramID(CAENRFIDHandle Handle, CAENRFIDTag *TagID, char Password, unsigned short Lock);

1.3.39. CAENRFID_KillTag

Name: CAENRFID_KillTag

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to kill an EPC Class 1 Gen 1 tag

Parameters: [in] Handle: The handle that identifies the device.
[in] TagID: The EPC of the tag.
[in] Password: The kill password for the tag.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_KillTag(CAENRFIDHandle Handle, CAENRFIDTag *TagID, char Password);

1.3.40. CAENRFID_BlockWrite

Name: CAENRFID_BlockWrite

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: This function allows to write Length bytes to the memory of a specific tag identified by the ID (regardless of its status) at the address specified by Address. This function doesn't work with semi-passive tags

Parameters: [in] Handle: The handle that identifies the device.
[in] ID: The tag ID.
[in] Address: The address of the memory to write.
[in] Length: The number of bytes to write.
[in] Data: The data to write in the tag's memory

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_BlockWrite(CAENRFIDHandle handle, CAENRFIDTag *ID, int Address, int Length, void *Data);

1.3.41. CAENRFID_SetRS232

Name: CAENRFID_SetRS232

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to configure the serial communication of the reader

Parameters: [in] Handle: The handle that identifies the device.
[in] baud: The baudrate value.
[in] datab: The databit value.
[in] stopb: The stopbit value.
[in] parity: The parity value.
[in] flowc: The flowcontrol value

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetRS232(CAENRFIDHandle handle, unsigned long baud, unsigned long datab, unsigned long stopb, CAENRFID_RS232_Parity parity, CAENRFID_RS232_FlowControl flowc);

1.3.42. CAENRFID_SetDateTime

Name: CAENRFID_SetDateTime

Reader: A828EU, A828US, A829EU, A829US, A946EU, A949EU, A928EU, A948EU

Description: The function permits to set the date e the time in the reader.

Parameters: [in] Handle: The handle that identifies the device.
[in] datetime: The current date ed time.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetDateTime(CAENRFIDHandle handle, char *datetime);
CAENRFIDlib_API CAENRFIDErrorCodes __stdcall _CAENRFID_GroupSelUnsel(CAENRFIDHandle handle, char *SourceName, CAENRFID_SelUnsel_Op code, int Address, int BitMask, void *data, CAENRFIDTag *ID);

1.3.43. CAENRFID_GetIO

Name: CAENRFID_GetIO

Reader: A928EU, A948EU

Description: The function permits to read the IO register

Parameters: [in] Handle: The handle that identifies the device.
[out] IORegister: The current IO Register

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetIO(CAENRFIDHandle handle, unsigned int *IORegister);

1.3.44. CAENRFID_SetIO

Name: CAENRFID_SetIO

Reader: A928EU, A948EU

Description: The function permits to write the IO register

Parameters: [in] Handle: The handle that identifies the device.
[in] IORegister: The IO Register value.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetIO(CAENRFIDHandle handle, unsigned int IORegister);

1.3.45. CAENRFID_SetSourceConfiguration

Name: CAENRFID_SetSourceConfiguration

Reader: A928EU, A948EU

Description: The function permits to configure the Logical Source

Parameters: [in] Handle : The handle that identifies the device.
[in] SourceName: The Name of the Logical Source.

[in] parameter: The parameter of Logical Source to configure.
[in] value: The the value of the parameter.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetSourceConfiguration(CAENRFIDHandle handle, char *SourceName, CAENRFID_SOURCE_Parameter parameter, int value);

1.3.46. CAENRFID_GetSourceConfiguration

Name: CAENRFID_GetSourceConfiguration

Reader: A928EU, A948EU

Description: The function permits to get the value of the Logical Source configuration

Parameters: [in] Handle: The handle that identifies the device.
[in] SourceName: The Name of the Logical Source.
[in] parameter: The parameter of Logical Source to configure.
[out] value: The the value of the parameter

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetSourceConfiguration(CAENRFIDHandle handle, char *SourceName, CAENRFID_SOURCE_Parameter parameter, int *pvalue);

1.3.47. CAENRFID_GetAllocatedTriggers

Name: CAENRFID_GetAllocatedTriggers

Reader: A928EU, A948EU

Description: The function permits to get the allocated triggers

Parameters: [in] Handle: The handle that identifies the device.
[out] TriggerNum: The number of triggers allocated.
[out] Triggers: The Triggers's names of allocated triggers

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetAllocatedTriggers(CAENRFIDHandle handle, int *TriggerNum, char **Triggers);

1.3.48. CAENRFID_GetAllocatedChannels

Name: CAENRFID_GetAllocatedTriggers

Reader: A928EU, A948EU

Description: The function permits to get the allocated channels

Parameters: [in] Handle: The handle that identifies the device.
[out] ChannelNum: The number of channels allocated.
[out] Channels: The channels's names of allocated channels

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID GetAllocatedChannels(CAENRFIDHandle handle,

int *ChannelNum, char **Channels);

1.3.49. CAENRFID_SetEventMode

Name: CAENRFID_SetEventMode

Reader: A928EU, A948EU

Description: The function permits to set the Event Generation Mode of the reader

Parameters: [in] Handle: The handle that identifies the device.
[in] EMode: The Event Mode

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetEventMode(CAENRFIDHandle handle, CAENRFID_EventMode EMode);

1.3.50. CAENRFID_GetEventMode

Name: CAENRFID_GetEventMode

Reader: A928EU, A948EU

Description: The function permits to get the Event Generation Mode of the reader

Parameters: [in] Handle: The handle that identifies the device.
[out] EMode: The Event Mode of the reader

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetEventMode(CAENRFIDHandle handle, CAENRFID_EventMode *EMode);

1.3.51. CAENRFID_FirmwareUpgrade

Name: CAENRFID_FirmwareUpgrade

Reader: A928EU, A948EU

Description: The function permits to upgrade the reader's firmware

Parameters: [in] Handle: The handle that identifies the device.
[in] type: The kind of upgrading
[in] arg: The argument for the upgrading in the form '[tftpserver ip]:[filename]'

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_FirmwareUpgrade(CAENRFIDHandle handle, CAENRFID_FWUpgradeType type, char *arg);

1.3.52. CAENRFID_Lock_C1G2

Name: CAENRFID_Lock_C1G2

Reader: A928EU, A948EU

Description: This function allows to lockthe memory of a specific tag identified by the ID and by Payload

Parameters: [in] Handle : The handle that identifies the device.
[in] ID : The tag ID.
[in] Payload : The payload of the tag's memory.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Lock_C1G2(CAENRFIDHandle handle, CAENRFIDTag *ID, int payload);

1.3.53. CAENRFID_KillTag_C1G2

Name: CAENRFID_KillTag_C1G2

Reader: A928EU, A948EU

Description: The function permits to kill an EPC Class 1 Gen 2 tag

Parameters: [in] Handle : The handle that identifies the device.
[in] TagID : The EPC of the tag.
[in] Password : The password for the tag.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_KillTag_C1G2(CAENRFIDHandle handle, CAENRFIDTag *ID, int password);

1.3.54. CAENRFID_KillTag_C1G2

Name: CAENRFID_KillTag_C1G2

Reader: A928EU, A948EU

Description: The function permits to kill an EPC Class 1 Gen 2 tag

Parameters: [in] Handle : The handle that identifies the device.
[in] TagID : The EPC of the tag.
[in] Password : The password for the tag.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_KillTag_C1G2(CAENRFIDHandle handle, CAENRFIDTag *ID, int password);

1.3.55. CAENRFID_ProgramID_EPC119

Name: CAENRFID_ProgramID_EPC119

Reader: A928EU, A948EU

Description: The function permits to program an EPC 119 tag

Parameters: [in] Handle: The handle that identifies the device.
[in] ID: The actual ID of the tag.
[in] NewID: The new ID for the specified tag.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_ProgramID_EPC119(CAENRFIDHandle handle, CAENRFIDTag *ID, char *NewID);

1.3.56. CAENRFID_ProgramID_C1G2

Name: CAENRFID_ProgramID_C1G2

Reader: A928EU, A948EU

Description: The function permits to program an EPC Class 1 Gen 2 tag

Parameters:
[in] Handle: The handle that identifies the device.
[in] ID: The EPC to program in the tag.
[in] nsi: The NSI value for the EPC C1G2.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_ProgramID_C1G2(CAENRFIDHandle handle, CAENRFIDTag *ID, unsigned short nsi);

1.3.57. CAENRFID_Read_C1G2

Name: CAENRFID_Read_C1G2

Reader: A928EU, A948EU

Description: This function allows to read Length bytes from the bank memory, specified by membank, of a specific tag identified by the ID (regardless of its status) at the address specified by Address.

Parameters:
[in] Handle : The handle that identifies the device.
[in] ID : The tag ID.
[in] membank : The memory Bank of EPC C1G2 Tag
[in] Address : The address of the memory to read.
[in] Length : The number of bytes to read.
[out] Data : The data read from the tag's memory.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Read_C1G2(CAENRFIDHandle handle, CAENRFIDTag *ID, short membank, int Address, int Length, void *Data);

1.3.58. CAENRFID_Write_C1G2

Name: CAENRFID_Write_C1G2

Reader: A928EU, A948EU

Description: This function allows to write Length bytes to the bank memory, specified by membank, of a specific tag identified by the ID (regardless of its status) at the address specified by Address.

Parameters:
[in] Handle : The handle that identifies the device.
[in] ID : The tag ID.
[in] membank : The memory Bank of EPC C1G2 Tag
[in] Address : The address of the memory to write.
[in] Length : The number of bytes to write.
[in] Data : The data to write in the tag's memory.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_Write_C1G2(CAENRFIDHandle handle, CAENRFIDTag *ID, short membank,int Address, int Length, void

*Data);

1.3.59. CAENRFID_QueryTag_C1G2

Name: CAENRFID_QueryTag_C1G2

Reader: A928EU, A948EU

Description: The function permits to perform the Query command of C1G2 protocol

Parameters: [in] SourceName : The Name of the Logical Source.
[out] isPresent: A flag indicating if the tag answered at Query command

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_QueryTag_C1G2(CAENRFIDHandle handle, char *SourceName, short *isPresent);

1.3.60. CAENRFID_SetQ_C1G2

Name: CAENRFID_SetQ_C1G2

Reader: A928EU, A948EU

Description: The function permits to set the Q parameter of C1G2 protocol

Parameters: [in] Q : The value of Q.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_SetQ_C1G2(CAENRFIDHandle handle, int Q);

1.3.61. CAENRFID_GetQ_C1G2

Name: CAENRFID_GetQ_C1G2

Reader: A928EU, A948EU

Description: The function permits to get the Q parameter of C1G2 protocol

Parameters: [out] Q : The value of Q.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetQ_C1G2(CAENRFIDHandle handle, int *Q);

1.3.62. CAENRFID_GetReaderInfo

Name: CAENRFID_GetReaderInfo

Reader: A928EU, A948EU

Description: Permits to read the Model and the Serial number of the Reader

Parameters: [out] Model : Returns the model of the reader.
[out] SerialNum : Returns the Serial number of the reader.

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API CAENRFIDErrorCodes __stdcall CAENRFID_GetReaderInfo(CAENRFIDHandle handle,char *Model, char *SerialNum);

1.3.63. CAENRFID_FreeTagsMemory

Name: CAENRFID_FreeTagsMemory

Reader: A928EU, A948EU

Description: The function permits to free the allocated by CAENRFIDInventory

Parameters: [in] Tags : Reference to CAENRFIDTag obtained from CAENRFIDInventory

Returns: An error code about the execution of the function

Syntax: CAENRFIDlib_API void __stdcall CAENRFID_FreeTagsMemory(CAENRFIDTag **Tags)