



Bluebox
Embedded UHF M800, M900, M950
BlueboxLib Library Description

iDTRONIC GmbH
Ludwig-Reichling-Straße 4
67059 Ludwigshafen
Germany/Deutschland

Phone: +49 621 6690094-0
Fax: +49 621 6690094-9
E-Mail: info@idtronic.de
Web: idtronic.de

Issue 1.18
– 30. September 2025 –

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2025
Printed in Germany

Contents

1	Introduction	7
2	Library Description	8
2.1	Typedef – BLUEBOX_Handle	8
2.2	Enumerations	8
2.2.1	Bluebox_ErrorCodes	8
2.2.2	BLUEBOX_Output	9
2.2.3	BLUEBOX_Input	9
2.2.4	BLUEBOX_Antenna	9
2.2.5	BLUEBOX_TagType.....	9
2.2.6	BLUEBOX_ICODE_SLI_S_PasswordIdentifier	11
2.2.7	BLUEBOX_ICODE_SLI_S_ProtectionStatus.....	11
2.2.8	BLUEBOX_MIFARE_Key.....	12
2.2.9	BLUEBOX_ISO18K6C_Bank	12
2.2.10	BLUEBOX_ISO18K6C_PasswordPermission	13
2.2.11	BLUEBOX_ISO18K6C_MemoryPermission	13
2.2.12	BLUEBOX_Reader.....	13
2.3	Definitions.....	14
2.3.1	BLUEBOX_EM4305_ID_SIZE.....	14
2.3.2	BLUEBOX_T5557_ID_SIZE	14
2.3.3	BLUEBOX_Q5_ID_SIZE	14
2.3.4	BLUEBOX_HITAG1_ID_SIZE	14
2.3.5	BLUEBOX_HITAG1_PAGE_SIZE	14
2.3.6	BLUEBOX_HITAGS_ID_SIZE	14
2.3.7	BLUEBOX_HITAGS_PAGE_SIZE.....	14
2.3.8	BLUEBOX_TITAN_ID_SIZE	14
2.3.9	BLUEBOX_TITAN_PASSWORD_SIZE	15
2.3.10	BLUEBOX_TITAN_PAGE_SIZE	15
2.3.11	BLUEBOX_ISO15693_UID_SIZE	15
2.3.12	BLUEBOX_ICODE_SLI_S_RND_SIZE	15
2.3.13	BLUEBOX_ICODE_SLI_S_PWD_SIZE	15
2.3.14	BLUEBOX_MIFARE_MINI_UID_SIZE	15
2.3.15	BLUEBOX_MIFARE_1k_UID_SIZE	15
2.3.16	BLUEBOX_MIFARE_1k_BLOCK_SIZE	15
2.3.17	BLUEBOX_MIFARE_4k_UID_SIZE	15
2.3.18	BLUEBOX_MIFARE_4k_BLOCK_SIZE	16
2.3.19	BLUEBOX_MIFARE_UL_UID_SIZE.....	16
2.3.20	BLUEBOX_MIFARE_UL_BLOCK_SIZE	16
2.3.21	BLUEBOX_MIFARE_KEY_SIZE	16
2.3.22	BLUEBOX_MIFARE_DESFIRE_UID_SIZE.....	16
2.3.23	BLUEBOX_MIFARE_7BUID_2k_UID_SIZE	16
2.3.24	BLUEBOX_MIFARE_7BUID_4k_UID_SIZE	16
2.3.25	BLUEBOX_MIFARE_PLUS_2k_UID_SIZE	16
2.3.26	BLUEBOX_MIFARE_PLUS_4k_UID_SIZE	16
2.3.27	BLUEBOX_NTAG21x_UID_SIZE	17
2.3.28	BLUEBOX_NTAG21x_BLOCK_SIZE.....	17
2.3.29	BLUEBOX_SR176_UID_SIZE	17
2.3.30	BLUEBOX_SR176_BLOCK_SIZE	17
2.3.31	BLUEBOX_JCOS_UID_SIZE.....	17

2.3.32	BLUEBOX_PICOPASS_UID_SIZE	17
2.3.33	BLUEBOX_ISO18K6B_UID_SIZE	17
2.3.34	BLUEBOX_ISO18K6B_BLOCK_SIZE	17
2.3.35	BLUEBOX_ISO18K6C_UID_SIZE	17
2.3.36	BLUEBOX_ISO18K6C_BLOCK_SIZE	18
2.3.37	BLUEBOX_ISO18K6C_ACC_PWD_SIZE	18
2.3.38	BLUEBOX_ISO18K6C_KILL_PWD_SIZE	18
2.3.39	BLUEBOX_ACTIVE_UID_SIZE	18
2.4	Data Structures	18
2.4.1	BLUEBOX_Tag	18
2.4.2	BLUEBOX_Notify	18
2.4.3	BLUEBOX_ICODE_SLI_S_BlockProtectionStatus	19
2.4.4	BLUEBOX_Registration	19
2.5	Functions	19
2.5.1	BLUEBOX_GetSwRelease	19
2.5.2	BLUEBOX_Init	20
2.5.3	BLUEBOX_End	20
2.5.4	BLUEBOX_Open	20
2.5.5	BLUEBOX_Close	20
2.5.6	BLUEBOX_SetAddress	21
2.5.7	BLUEBOX_SetDevice	21
2.5.8	BLUEBOX_GetDevice	22
2.5.9	BLUEBOX_SetChannel	23
2.5.10	BLUEBOX_GetFwRelease	23
2.5.11	BLUEBOX_Reset	23
2.5.12	BLUEBOX_GetDateTime	24
2.5.13	BLUEBOX_SetDateTime	24
2.5.14	BLUEBOX_ReadParameters	24
2.5.15	BLUEBOX_WriteParameters	25
2.5.16	BLUEBOX_DefaultParameters	25
2.5.17	BLUEBOX_ReadSerialNumber	25
2.5.18	BLUEBOX_ReadMACAddress	26
2.5.19	BLUEBOX_ReadConfiguration	26
2.5.20	BLUEBOX_WriteConfiguration	26
2.5.21	BLUEBOX_DefaultConfiguration	27
2.5.22	BLUEBOX_DataRequest	27
2.5.23	BLUEBOX_QueueRequest	27
2.5.24	BLUEBOX_FreeTagsMemory	28
2.5.25	BLUEBOX_AllocateNotifyChannel	28
2.5.26	BLUEBOX_DeallocateNotifyChannel	28
2.5.27	BLUEBOX_GetNotification	28
2.5.28	BLUEBOX_FreeNotifyMemory	29
2.5.29	BLUEBOX_SetOutput	29
2.5.30	BLUEBOX_GetReaderStatus	29
2.5.31	BLUEBOX_GetTemperature	30
2.5.32	BLUEBOX_RfOnOff	30
2.5.33	BLUEBOX_SelectiveRfOnOff	30
2.5.34	BLUEBOX_ReadID_EM4305	31
2.5.35	BLUEBOX_Write_EM4305	31
2.5.36	BLUEBOX_ReadID_T5557	32

2.5.37	BLUEBOX_Write_T5557	32
2.5.38	BLUEBOX_ReadID_Q5	33
2.5.39	BLUEBOX_Write_Q5	33
2.5.40	BLUEBOX_ReadID_HITAG1	34
2.5.41	BLUEBOX_ReadPage_HITAG1	34
2.5.42	BLUEBOX_WritePage_HITAG1	35
2.5.43	BLUEBOX_ReadID_HITAGS	35
2.5.44	BLUEBOX_Write_HITAGS	36
2.5.45	BLUEBOX_ReadPage_HITAGS	36
2.5.46	BLUEBOX_WritePage_HITAGS	37
2.5.47	BLUEBOX_ReadID_TITAN	37
2.5.48	BLUEBOX_Reset_TITAN	38
2.5.49	BLUEBOX_Login_TITAN	38
2.5.50	BLUEBOX_WritePassword_TITAN	39
2.5.51	BLUEBOX_SelectiveRead_TITAN	39
2.5.52	BLUEBOX_SelectiveWrite_TITAN	40
2.5.53	BLUEBOX_Inventory_ISO15693	40
2.5.54	BLUEBOX_ReadPage_ISO15693	41
2.5.55	BLUEBOX_ReadMultiPage_ISO15693	42
2.5.56	BLUEBOX_WritePage_ISO15693	42
2.5.57	BLUEBOX_WriteMultiPage_ISO15693	43
2.5.58	BLUEBOX_LockPage_ISO15693	43
2.5.59	BLUEBOX_Write_AFI_ISO15693	44
2.5.60	BLUEBOX_Lock_AFI_ISO15693	44
2.5.61	BLUEBOX_GetRandomNumber_ICODE_SLI_S	45
2.5.62	BLUEBOX_SetPassword_ICODE_SLI_S	45
2.5.63	BLUEBOX_WritePassword_ICODE_SLI_S	46
2.5.64	BLUEBOX_LockPassword_ICODE_SLI_S	46
2.5.65	BLUEBOX_64BitPasswordProtection_ICODE_SLI_S	47
2.5.66	BLUEBOX_ProtectPage_ICODE_SLI_S	47
2.5.67	BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S	48
2.5.68	BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S	48
2.5.69	BLUEBOX_Destroy_SLI_S_ICODE_SLI_S	49
2.5.70	BLUEBOX_EnablePrivacy_ICODE_SLI_S	49
2.5.71	BLUEBOX_Inventory_ISO14443A	50
2.5.72	BLUEBOX_ReadBlock_MIFARE_1k	50
2.5.73	BLUEBOX_WriteBlock_MIFARE_1k	51
2.5.74	BLUEBOX_ReadBlock_MIFARE_4k	52
2.5.75	BLUEBOX_WriteBlock_MIFARE_4k	52
2.5.76	BLUEBOX_ReadBlock_MIFARE_Ultralight	53
2.5.77	BLUEBOX_WriteBlock_MIFARE_Ultralight	53
2.5.78	BLUEBOX_ReadBlock_NTAG213	54
2.5.79	BLUEBOX_WriteBlock_NTAG213	54
2.5.80	BLUEBOX_ReadBlock_NTAG215	55
2.5.81	BLUEBOX_WriteBlock_NTAG215	56
2.5.82	BLUEBOX_ReadBlock_NTAG216	56
2.5.83	BLUEBOX_WriteBlock_NTAG216	57
2.5.84	BLUEBOX_Inventory_ISO14443B	57
2.5.85	BLUEBOX_ReadBlock_SR176	58
2.5.86	BLUEBOX_WriteBlock_SR176	58

2.5.87	BLUEBOX_Inventory_PICOPASS	59
2.5.88	BLUEBOX_ReadRfParameters	59
2.5.89	BLUEBOX_WriteRfParameters	60
2.5.90	BLUEBOX_Inventory_ISO18K6B	60
2.5.91	BLUEBOX_Read_ISO18K6B.....	60
2.5.92	BLUEBOX_Write_ISO18K6B.....	61
2.5.93	BLUEBOX_Inventory_ISO18K6C	61
2.5.94	BLUEBOX_ProgramEPC_ISO18K6C.....	62
2.5.95	BLUEBOX_Read_ISO18K6C.....	62
2.5.96	BLUEBOX_Write_ISO18K6C.....	63
2.5.97	BLUEBOX_BlockWrite_ISO18K6C.....	63
2.5.98	BLUEBOX_Lock_ISO18K6C.....	64
2.5.99	Bluebox_Kill_ISO18K6C	66
2.5.100	BLUEBOX_FwUpgrade	66
2.5.101	BLUEBOX_ReadNumberOfRegistrations	66
2.5.102	BLUEBOX_ReadOlderRegistration	67
2.5.103	BLUEBOX_CancelOlderRegistration	67
2.5.104	BLUEBOX_CancelAllRegistrations.....	67
2.5.105	BLUEBOX_ReadPreviousRegistration	68
2.5.106	BLUEBOX_GenericCommand	68
3	BlueBox Gen1 Functions Table	69
4	BlueBox Gen2 Functions Table	73
5	BlueBox CX Functions Table	77
6	Document Revision History	80

1 Introduction

This manual describes the BlueboxLib library and its implemented functions.

BlueboxLib is a set of ANSI C functions which allows a user program to use and configure the Bluebox and embedded UHF devices. The library is available in the formats Win32 DLL and x64 DLL.

2 Library Description

2.1 Typedef – BLUEBOX_Handle

Name	BLUEBOX_HANDLE
Description	Handle type used to identify readers
Syntax	<code>typedef int BLUEBOX_Handle;</code>

2.2 Enumerations

2.2.1 Bluebox_ErrorCodes

Name	BLUEBOX_ErrorCodes	
Description	Error codes enum	
Enumerator	BLUEBOX_StatusOk	Operation completed successfully
	BLUEBOX_InitError	Initialization error
	BLUEBOX_InvalidHandle	Invalid handle
	BLUEBOX_InvalidChannel	Invalid channel
	BLUEBOX_InvalidParams	Invalid parameters
	BLUEBOX_GenericError	Generic error
	BLUEBOX_TimeoutError	Timeout error
	BLUEBOX_CommunicationError	Communication error
	BLUEBOX_ConnectionError	Connection error
	BLUEBOX_MemoryError	Memory allocation error
	BLUEBOX_InvalidCommand	Invalid command
	BLUEBOX_TagNotFound	Tag not found
	BLUEBOX_TagError	Tag error
	BLUEBOX_AllocationError	Notify channel allocation error
	BLUEBOX_FileError	File error
	BLUEBOX_RegistrationNotFound	Registration not found
Syntax	<pre>typedef enum BLUEBOX_ErrorCodes { BLUEBOX_StatusOk = 0, BLUEBOX_InitError = -1, BLUEBOX_InvalidHandle = -2, BLUEBOX_InvalidChannel = -3, BLUEBOX_InvalidParams = -4, BLUEBOX_GenericError = -5, BLUEBOX_TimeoutError = -6, BLUEBOX_CommunicationError = -7, BLUEBOX_ConnectionError = -8, BLUEBOX_MemoryError = -9, BLUEBOX_InvalidCommand = -10, BLUEBOX_TagNotFound = -11, BLUEBOX_TagError = -12, BLUEBOX_AllocationError = -13, BLUEBOX_FileError = -14, BLUEBOX_RegistrationNotFound = -15 } BLUEBOX_ErrorCodes;</pre>	

2.2.2 BLUEBOX_Output

Name	BLUEBOX_Output	
Description	Output enum	
Enumerator	BLUEBOX_OUTPUT_1	Output 1
	BLUEBOX_OUTPUT_2	Output 2
Syntax	<pre>typedef enum BLUEBOX_Output { BLUEBOX_OUTPUT_1 = 1, BLUEBOX_OUTPUT_2 = 2 } BLUEBOX_Output;</pre>	

2.2.3 BLUEBOX_Input

Name	BLUEBOX_Input	
Description	Input enum	
Enumerator	BLUEBOX_NOINPUT	No input information
	BLUEBOX_INPUT_1	Input 1
	BLUEBOX_INPUT_2	Input 2
Syntax	<pre>typedef enum BLUEBOX_Input { BLUEBOX_NOINPUT = 0, BLUEBOX_INPUT_1 = 1, BLUEBOX_INPUT_2 = 2 } BLUEBOX_Input;</pre>	

2.2.4 BLUEBOX_Antenna

Name	BLUEBOX_Antenna	
Description	Antenna enum	
Enumerator	BLUEBOX_NOANT	No antenna information
	BLUEBOX_ANT_1	Antenna no. 1
	BLUEBOX_ANT_2	Antenna no. 2
	BLUEBOX_ANT_3	Antenna no. 3
	BLUEBOX_ANT_4	Antenna no. 4
Syntax	<pre>typedef enum BLUEBOX_Antenna { BLUEBOX_NOANT = 0, BLUEBOX_ANT_1 = 1, BLUEBOX_ANT_2 = 2, BLUEBOX_ANT_3 = 3, BLUEBOX_ANT_4 = 4 } BLUEBOX_Antenna;</pre>	

2.2.5 BLUEBOX_TagType

Name	BLUEBOX_TagType	
Description	Tag type enum	
Enumerator	BLUEBOX_UNDEFINED	Undefined tag
	BLUEBOX_SHORT	BLUEBOX SHORT
	BLUEBOX_MEDIUM	BLUEBOX MEDIUM
	BLUEBOX_LARGE	BLUEBOX LARGE
	BLUEBOX_Q5	Q5
	BLUEBOX_HITAG_S256	HITAG S 256

	BLUEBOX_HITAG_S2048	HITAG S 2048
	BLUEBOX_TITAN	TITAN
	BLUEBOX_ISO14443A	ISO 14443A
	BLUEBOX_MIFARE_1k	MIFARE 1k
	BLUEBOX_MIFARE_4k	MIFARE 4k
	BLUEBOX_MIFARE_UL	MIFARE Ultralight
	BLUEBOX_ISO15693	ISO 15693
	BLUEBOX_ICODE2	ICODE SLI
	BLUEBOX_TAG_IT_HF_I	Tag-It HF-I
	BLUEBOX_EM4035	EM4035
	BLUEBOX_LRI_64_512	LRI 64/512
	BLUEBOX_MB89R118	MB89R118
	BLUEBOX_ISO14443B	ISO 14443B
	BLUEBOX_SR176	SR176
	BLUEBOX_ISO18K6B	ISO 18000-6B
	BLUEBOX_ISO18K6C	ISO 18000-6C
	BLUEBOX_ACTIVE	ACTIVE
	BLUEBOX_EM4305	EM4305
	BLUEBOX_T5557	T5557
	BLUEBOX_ICODE_SLI_S	ICODE SLI-S
	BLUEBOX_HITAG_1	HITAG 1
	BLUEBOX_MIFARE_MINI	MIFARE Mini
	BLUEBOX_MIFARE_DESFIRE	MIFARE Desfire
	BLUEBOX_MIFARE_7BUID_2k	MIFARE 2k with 7 bytes UID
	BLUEBOX_MIFARE_7BUID_4k	MIFARE 4k with 7 bytes UID
	BLUEBOX_MIFARE_PLUS_2k	MIFARE Plus 2k
	BLUEBOX_MIFARE_PLUS_4k	MIFARE Plus 4k
	BLUEBOX_SRI512	SRI 512
	BLUEBOX_JCOS	JCos
	BLUEBOX_PICOPASS	Picopass
	BLUEBOX_HITAG_2	HITAG 2
	BLUEBOX_JCOP	JCOP
	BLUEBOX_JCOP31	JCOP 31
	BLUEBOX_JCOP41	JCOP 41
Syntax	<pre>typedef enum BLUEBOX_TagType { BLUEBOX_UNDEFINED = 0, BLUEBOX_SHORT = 1, BLUEBOX_MEDIUM = 2, BLUEBOX_LARGE = 3, BLUEBOX_Q5 = 4, BLUEBOX_HITAG_S256 = 5, BLUEBOX_HITAG_S2048 = 6, BLUEBOX_TITAN = 7, BLUEBOX_ISO14443A = 8, BLUEBOX_MIFARE_1k = 9, BLUEBOX_MIFARE_4k = 10, BLUEBOX_MIFARE_UL = 11, BLUEBOX_ISO15693 = 12, BLUEBOX_ICODE2 = 13,</pre>	

```

BLUEBOX_TAG_IT_HF_I = 14,
BLUEBOX_EM4035 = 15,
BLUEBOX_LRI_64_512 = 16,
BLUEBOX_MB89R118 = 17,
BLUEBOX_ISO14443B = 18,
BLUEBOX_SR176 = 19,
BLUEBOX_ISO18K6B = 20,
BLUEBOX_ISO18K6C = 21,
BLUEBOX_ACTIVE = 22,
BLUEBOX_EM4305 = 23,
BLUEBOX_T5557 = 24,
BLUEBOX_ICODE_SLI_S = 25,
BLUEBOX_HITAG_1 = 26,
BLUEBOX_MIFARE_MINI = 27,
BLUEBOX_MIFARE_DESFIRE = 28,
BLUEBOX_MIFARE_7BUID_2k = 29,
BLUEBOX_MIFARE_7BUID_4k = 30,
BLUEBOX_MIFARE_PLUS_2k = 31,
BLUEBOX_MIFARE_PLUS_4k = 32,
BLUEBOX_SRI512 = 33,
BLUEBOX_JCOS = 34,
BLUEBOX_PICOPASS = 35,
BLUEBOX_HITAG_2 = 36,
BLUEBOX_JCOP = 37,
BLUEBOX_JCOP31 = 38,
BLUEBOX_JCOP41 = 39
} BLUEBOX_TagType;

```

2.2.6 BLUEBOX_ICODE_SLI_S_PasswordIdentifier

Name	BLUEBOX_ICODE_SLI_S_PasswordIdentifier	
Description	ICODE SLI-S password identifier enum	
Enumerator	BLUEBOX_ICODE_SLI_S_PWD_READ	Read
	BLUEBOX_ICODE_SLI_S_PWD_WRITE	Write
	BLUEBOX_ICODE_SLI_S_PWD_PRIVACY	Privacy
	BLUEBOX_ICODE_SLI_S_PWD_DESTROY_SLI_S	Destroy SLI-S
	BLUEBOX_ICODE_SLI_S_PWD_EAS	EAS
Syntax	<pre> typedef enum BLUEBOX_ICODE_SLI_S_PasswordIdentifier { BLUEBOX_ICODE_SLI_S_PWD_READ = 0x01, BLUEBOX_ICODE_SLI_S_PWD_WRITE = 0x02, BLUEBOX_ICODE_SLI_S_PWD_PRIVACY = 0x04, BLUEBOX_ICODE_SLI_S_PWD_DESTROY_SLI_S = 0x08, BLUEBOX_ICODE_SLI_S_PWD_EAS = 0x10 } BLUEBOX_ICODE_SLI_S_PasswordIdentifier; </pre>	

2.2.7 BLUEBOX_ICODE_SLI_S_ProtectionStatus

Name	BLUEBOX_ICODE_SLI_S_ProtectionStatus	
Description	ICODE SLI-S protection status enum	
Enumerator	BLUEBOX_ICODE_SLI_S_PROTECT_PUBLIC	Public

	BLUEBOX_ICODE_SLI_S_PROTECT_READ_AND_WRITE_BY_READ_PWD	Protect Read and Write by Read password (32 bit)
	BLUEBOX_ICODE_SLI_S_PROTECT_WRITE_BY_WRITE_PWD	Protect Write by Write password (32 bit)
	BLUEBOX_ICODE_SLI_S_PROTECT_READ_BY_READ_PWD_AND_WRITE_BY_WRITE_PWD	Protect Read by Read password and Write by Write password (32bit)
	BLUEBOX_ICODE_SLI_S_PROTECT_READ_AND_WRITE_BY_READ_AND_WRITE_PWD	Protect Read and Write by Read and Write password (64bit)
	BLUEBOX_ICODE_SLI_S_PROTECT_WRITE_BY_READ_AND_WRITE_PWD	Protect Write by Read and Write password (64 bit)
Syntax	<pre>typedef enum BLUEBOX_ICODE_SLI_S_ProtectionStatus { BLUEBOX_ICODE_SLI_S_PROTECT_PUBLIC = 0x00, BLUEBOX_ICODE_SLI_S_PROTECT_READ_AND_WRITE_BY_READ_PWD = 0x01, BLUEBOX_ICODE_SLI_S_PROTECT_WRITE_BY_WRITE_PWD = 0x10, BLUEBOX_ICODE_SLI_S_PROTECT_READ_BY_READ_PWD_AND_WRITE_BY_WRITE_PWD = 0x11, BLUEBOX_ICODE_SLI_S_PROTECT_READ_AND_WRITE_BY_READ_AND_WRITE_PWD = 0x01, BLUEBOX_ICODE_SLI_S_PROTECT_WRITE_BY_READ_AND_WRITE_PWD = 0x10, } BLUEBOX_ICODE_SLI_S_ProtectionStatus;</pre>	

2.2.8 BLUEBOX_MIFARE_Key

Name	BLUEBOX_MIFARE_Key	
Description	MIFARE key enum	
Enumerator	BLUEBOX_MIFARE_KEY_A	Key A
	BLUEBOX_MIFARE_KEY_B	Key B
Syntax	<pre>typedef enum BLUEBOX_MIFARE_Key { BLUEBOX_MIFARE_KEY_A = 0, BLUEBOX_MIFARE_KEY_B = 1 } BLUEBOX_MIFARE_Key;</pre>	

2.2.9 BLUEBOX_ISO18K6C_Bank

Name	BLUEBOX_ISO18K6C_Bank	
Description	ISO18000-6B tag's memory bank enum	
Enumerator	BLUEBOX_ISO18K6C_BANK_RESERVED	Reserved
	BLUEBOX_ISO18K6C_BANK_EPC	EPC
	BLUEBOX_ISO18K6C_BANK_TID	TID
	BLUEBOX_ISO18K6C_BANK_USER	User
Syntax	<pre>typedef enum BLUEBOX_ISO18K6C_Bank { BLUEBOX_ISO18K6C_BANK_RESERVED = 0, BLUEBOX_ISO18K6C_BANK_EPC = 1, BLUEBOX_ISO18K6C_BANK_TID = 2, BLUEBOX_ISO18K6C_BANK_USER = 3 } BLUEBOX_ISO18K6C_Bank;</pre>	

2.2.10 BLUEBOX_ISO18K6C_PasswordPermission

Name	BLUEBOX_ISO18K6C_PasswordPermission	
Description	The ISO18000-6C tag password permission values enum	
Enumerator	BLUEBOX_ISO18K6C_TAG_PWD_PERM_ACCESSIBLE	Accessible from either opened and secured states
	BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_ACCESSIBLE	Permanently accessible from either opened and secured states. It couldn't be locked
	BLUEBOX_ISO18K6C_TAG_PWD_PERM_SECURED_ACCESSIBLE	Accessible only from secured state
	BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_NOT_ACCESSIBLE	Not accessible from either opened or secured states
	BLUEBOX_ISO18K6C_TAG_PWD_PERM_NO_CHANGE	No change in accessible options
Syntax	<pre>typedef enum BLUEBOX_ISO18K6C_PasswordPermission { BLUEBOX_ISO18K6C_TAG_PWD_PERM_ACCESSIBLE = 0, BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_ACCESSIBLE = 1, BLUEBOX_ISO18K6C_TAG_PWD_PERM_SECURED_ACCESSIBLE = 2, BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_NOT_ACCESSIBLE = 3, BLUEBOX_ISO18K6C_TAG_PWD_PERM_NO_CHANGE = 4 } BLUEBOX_ISO18K6C_PasswordPermission;</pre>	

2.2.11 BLUEBOX_ISO18K6C_MemoryPermission

Name	BLUEBOX_ISO18K6C_MemoryPermission	
Description	The ISO18000-6C tag memory permission values enum	
Enumerator	BLUEBOX_ISO18K6C_TAG_MEM_PERM_WRITABLE	Writable from either opened and secured states
	BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_WRITABLE	Permanently writable from either opened and secured states. It can't be locked
	BLUEBOX_ISO18K6C_TAG_MEM_PERM_SECURED_WRITABLE	Writable only from secured state
	BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_NOT_WRITABLE	Not writable from either opened or secured states
	BLUEBOX_ISO18K6C_TAG_MEM_PERM_NO_CHANGE	No change in writable options
Syntax	<pre>typedef enum BLUEBOX_ISO18K6C_MemoryPermission { BLUEBOX_ISO18K6C_TAG_MEM_PERM_WRITABLE = 0, BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_WRITABLE = 1, BLUEBOX_ISO18K6C_TAG_MEM_PERM_SECURED_WRITABLE = 2, BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_NOT_WRITABLE = 3, BLUEBOX_ISO18K6C_TAG_MEM_PERM_NO_CHANGE = 4 } BLUEBOX_ISO18K6C_MemoryPermission;</pre>	

2.2.12 BLUEBOX_Reader

Name	BLUEBOX_Reader	
Description	Reader (primary, auxiliary,..) ID	
Enumerator	BLUEBOX_PRIMARY_READER	Primary reader
	BLUEBOX_AUXILIARY_1_READER	First auxiliary reader
	BLUEBOX_AUXILIARY_2_READER	2nd auxiliary reader

Syntax	<pre>typedef enum BLUEBOX_Reader { BLUEBOX_PRIMARY_READER = 0, BLUEBOX_AUXILIARY_1_READER = 1, BLUEBOX_AUXILIARY_2_READER = 2, } BLUEBOX_Reader;</pre>
---------------	--

2.3 Definitions

2.3.1 BLUEBOX_EM4305_ID_SIZE

Name	BLUEBOX_EM4305_ID_SIZE
Description	EM4305 tag's ID size in bytes
Syntax	<code>#define BLUEBOX_EM4305_ID_SIZE (4)</code>

2.3.2 BLUEBOX_T5557_ID_SIZE

Name	BLUEBOX_T5557_ID_SIZE
Description	T5557 tag's ID size in bytes
Syntax	<code>#define BLUEBOX_T5557_ID_SIZE (8)</code>

2.3.3 BLUEBOX_Q5_ID_SIZE

Name	BLUEBOX_Q5_ID_SIZE
Description	Q5 tag's ID size in bytes
Syntax	<code>#define BLUEBOX_Q5_ID_SIZE (5)</code>

2.3.4 BLUEBOX_HITAG1_ID_SIZE

Name	BLUEBOX_HITAG1_ID_SIZE
Description	HITAG 1 tag's ID size in bytes
Syntax	<code>#define BLUEBOX_HITAG1_ID_SIZE (4)</code>

2.3.5 BLUEBOX_HITAG1_PAGE_SIZE

Name	BLUEBOX_HITAG1_PAGE_SIZE
Description	HITAG 1 tag's memory page size in bytes
Syntax	<code>#define BLUEBOX_HITAG1_PAGE_SIZE (4)</code>

2.3.6 BLUEBOX_HITAGS_ID_SIZE

Name	BLUEBOX_HITAGS_ID_SIZE
Description	HITAG S tag's ID size in bytes
Syntax	<code>#define BLUEBOX_HITAGS_ID_SIZE (4)</code>

2.3.7 BLUEBOX_HITAGS_PAGE_SIZE

Name	BLUEBOX_HITAGS_PAGE_SIZE
Description	HITAG S tag's memory page size in bytes
Syntax	<code>#define BLUEBOX_HITAGS_PAGE_SIZE (4)</code>

2.3.8 BLUEBOX_TITAN_ID_SIZE

Name	BLUEBOX_TITAN_ID_SIZE
-------------	-----------------------

Description	TITAN tag's ID size in bytes
Syntax	<code>#define BLUEBOX_TITAN_ID_SIZE (8)</code>

2.3.9 BLUEBOX_TITAN_PASSWORD_SIZE

Name	BLUEBOX_TITAN_PASSWORD_SIZE
Description	TITAN tag's password size in bytes
Syntax	<code>#define BLUEBOX_TITAN_PASSWORD_SIZE (4)</code>

2.3.10 BLUEBOX_TITAN_PAGE_SIZE

Name	BLUEBOX_TITAN_PAGE_SIZE
Description	TITAN tag's page size in bytes
Syntax	<code>#define BLUEBOX_TITAN_PAGE_SIZE (4)</code>

2.3.11 BLUEBOX_ISO15693_UID_SIZE

Name	BLUEBOX_ISO15693_UID_SIZE
Description	ISO15693 tag's UID size in bytes
Syntax	<code>#define BLUEBOX_ISO15693_UID_SIZE (8)</code>

2.3.12 BLUEBOX_ICODE_SLI_S_RND_SIZE

Name	BLUEBOX_ICODE_SLI_S_RND_SIZE
Description	ICODE SLI-S tag's random number size in bytes
Syntax	<code>#define BLUEBOX_ICODE_SLI_S_RND_SIZE (2)</code>

2.3.13 BLUEBOX_ICODE_SLI_S_PWD_SIZE

Name	BLUEBOX_ICODE_SLI_S_PWD_SIZE
Description	ICODE SLI-S tag's password size in bytes
Syntax	<code>#define BLUEBOX_ICODE_SLI_S_PWD_SIZE (4)</code>

2.3.14 BLUEBOX_MIFARE_MINI_UID_SIZE

Name	BLUEBOX_MIFARE_MINI_UID_SIZE
Description	MIFARE Mini tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_MINI_UID_SIZE (4)</code>

2.3.15 BLUEBOX_MIFARE_1k_UID_SIZE

Name	BLUEBOX_MIFARE_1k_UID_SIZE
Description	MIFARE 1k tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_1k_UID_SIZE (4)</code>

2.3.16 BLUEBOX_MIFARE_1k_BLOCK_SIZE

Name	BLUEBOX_MIFARE_1k_BLOCK_SIZE
Description	MIFARE 1k tag's memory block size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_1k_BLOCK_SIZE (16)</code>

2.3.17 BLUEBOX_MIFARE_4k_UID_SIZE

Name	BLUEBOX_MIFARE_4k_UID_SIZE
-------------	----------------------------

Description	MIFARE 4k tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_4k_UID_SIZE (4)</code>

2.3.18 BLUEBOX_MIFARE_4k_BLOCK_SIZE

Name	BLUEBOX_MIFARE_4k_BLOCK_SIZE
Description	MIFARE 4k tag's memory block size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_4k_BLOCK_SIZE (16)</code>

2.3.19 BLUEBOX_MIFARE_UL_UID_SIZE

Name	BLUEBOX_MIFARE_UL_UID_SIZE
Description	MIFARE Ultralight tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_UL_UID_SIZE (7)</code>

2.3.20 BLUEBOX_MIFARE_UL_BLOCK_SIZE

Name	BLUEBOX_MIFARE_UL_BLOCK_SIZE
Description	MIFARE Ultralight tag's memory block size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_UL_BLOCK_SIZE (4)</code>

2.3.21 BLUEBOX_MIFARE_KEY_SIZE

Name	BLUEBOX_MIFARE_KEY_SIZE
Description	MIFARE tag's key size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_KEY_SIZE (6)</code>

2.3.22 BLUEBOX_MIFARE_DESFIRE_UID_SIZE

Name	BLUEBOX_MIFARE_DESFIRE_UID_SIZE
Description	MIFARE DESFire tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_DESFIRE_UID_SIZE (7)</code>

2.3.23 BLUEBOX_MIFARE_7BUID_2k_UID_SIZE

Name	BLUEBOX_MIFARE_7BUID_2k_UID_SIZE
Description	MIFARE 2k 7-bytes UID tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_7BUID_2k_UID_SIZE (7)</code>

2.3.24 BLUEBOX_MIFARE_7BUID_4k_UID_SIZE

Name	BLUEBOX_MIFARE_7BUID_4k_UID_SIZE
Description	MIFARE 4k 7-bytes UID tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_7BUID_4k_UID_SIZE (7)</code>

2.3.25 BLUEBOX_MIFARE_PLUS_2k_UID_SIZE

Name	BLUEBOX_MIFARE_PLUS_2k_UID_SIZE
Description	MIFARE Plus 2k tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_PLUS_2k_UID_SIZE (7)</code>

2.3.26 BLUEBOX_MIFARE_PLUS_4k_UID_SIZE

Name	BLUEBOX_MIFARE_PLUS_4k_UID_SIZE
-------------	---------------------------------

Description	MIFARE Plus 4k tag's UID size in bytes
Syntax	<code>#define BLUEBOX_MIFARE_PLUS_4k_UID_SIZE (7)</code>

2.3.27 BLUEBOX_NTAG21x_UID_SIZE

Name	BLUEBOX_NTAG21x_UID_SIZE
Description	NTAG21x tag's UID size in bytes
Syntax	<code>#define BLUEBOX_NTAG21x_UID_SIZE (7)</code>

2.3.28 BLUEBOX_NTAG21x_BLOCK_SIZE

Name	BLUEBOX_NTAG21x_UID_SIZE
Description	NTAG21x tag's UID size in bytes
Syntax	<code>#define BLUEBOX_NTAG21x_UID_SIZE (7)</code>

2.3.29 BLUEBOX_SR176_UID_SIZE

Name	BLUEBOX_SR176_UID_SIZE
Description	SR176 tag's UID size in bytes
Syntax	<code>#define BLUEBOX_SR176_UID_SIZE (8)</code>

2.3.30 BLUEBOX_SR176_BLOCK_SIZE

Name	BLUEBOX_SR176_BLOCK_SIZE
Description	SR176 tag's memory block size in bytes
Syntax	<code>#define BLUEBOX_SR176_BLOCK_SIZE (2)</code>

2.3.31 BLUEBOX_JCOS_UID_SIZE

Name	BLUEBOX_JCOS_UID_SIZE
Description	JCos tag's UID size in bytes
Syntax	<code>#define BLUEBOX_JCOS_UID_SIZE (8)</code>

2.3.32 BLUEBOX_PICOPASS_UID_SIZE

Name	BLUEBOX_PICOPASS_UID_SIZE
Description	Picopass tag's UID size in bytes
Syntax	<code>#define BLUEBOX_PICOPASS_UID_SIZE (8)</code>

2.3.33 BLUEBOX_ISO18K6B_UID_SIZE

Name	BLUEBOX_ISO18K6B_UID_SIZE
Description	ISO18000-6B tag's UID size in bytes
Syntax	<code>#define BLUEBOX_ISO18K6B_UID_SIZE (8)</code>

2.3.34 BLUEBOX_ISO18K6B_BLOCK_SIZE

Name	BLUEBOX_ISO18K6B_BLOCK_SIZE
Description	ISO18000-6B tag's memory block size in bytes
Syntax	<code>#define BLUEBOX_ISO18K6B_BLOCK_SIZE (8)</code>

2.3.35 BLUEBOX_ISO18K6C_UID_SIZE

Name	BLUEBOX_ISO18K6C_UID_SIZE
-------------	---------------------------

Description	ISO18000-6C tag's UID maximum size in bytes
Syntax	<code>#define BLUEBOX_ISO18K6C_UID_SIZE (66)</code>

2.3.36 BLUEBOX_ISO18K6C_BLOCK_SIZE

Name	BLUEBOX_ISO18K6C_BLOCK_SIZE
Description	ISO18000-6C tag's memory block size in bytes
Syntax	<code>#define BLUEBOX_ISO18K6C_BLOCK_SIZE (2)</code>

2.3.37 BLUEBOX_ISO18K6C_ACC_PWD_SIZE

Name	BLUEBOX_ISO18K6C_ACC_PWD_SIZE
Description	ISO18000-6C tag's access password size in bytes
Syntax	<code>#define BLUEBOX_ISO18K6C_ACC_PWD_SIZE (4)</code>

2.3.38 BLUEBOX_ISO18K6C_KILL_PWD_SIZE

Name	BLUEBOX_ISO18K6C_KILL_PWD_SIZE
Description	ISO18000-6C tag's kill password size in bytes
Syntax	<code>#define BLUEBOX_ISO18K6C_KILL_PWD_SIZE (4)</code>

2.3.39 BLUEBOX_ACTIVE_UID_SIZE

Name	BLUEBOX_ACTIVE_UID_SIZE
Description	ACTIVE tag's UID size in bytes
Syntax	<code>#define BLUEBOX_ACTIVE_UID_SIZE (8)</code>

2.4 Data Structures

2.4.1 BLUEBOX_Tag

Name	BLUEBOX_Tag	
Description	Tag identification struct	
Data fields	TagType	Tag type
	Id	Pointer to tag ID
	Length	The length of the tag ID in bytes
	Antenna	The antenna that identifies the tag
	Input	The input (direction) of tag identification
Syntax	<pre>typedef struct BLUEBOX_Tag { BLUEBOX_TagType TagType; unsigned char* Id; int Length; BLUEBOX_Antenna Antenna; BLUEBOX_Input Input; } BLUEBOX_Tag;</pre>	

2.4.2 BLUEBOX_Notify

Name	BLUEBOX_Notify	
Description	Tag notification struct	
Data fields	Address	The address of the reader that identifies the tag
	TagType	Tag type

	Id	Pointer to the tag ID
	Length	The length of the tag ID in bytes
	Antenna	The antenna which identified the tag
	Input	The input (direction) of tag identification
Syntax	<pre>typedef struct BLUEBOX_Notify { unsigned char Address; BLUEBOX_TagType TagType; unsigned char* Id; int Length; BLUEBOX_Antenna Antenna; BLUEBOX_Input Input; } BLUEBOX_Notify;</pre>	

2.4.3 BLUEBOX_ICODE_SLI_S_BlockProtectionStatus

Name	BLUEBOX_ICODE_SLI_S_BlockProtectionStatus	
Description	ICODE SLI-S block protection status struct	
Data fields	Lock	Lock bit (write access condition)
	ReadPasswordProtected	Read password protected
	WritePasswordProtected	Write password protected
	PageProtectionLock	Page protection lock
Syntax	<pre>typedef struct BLUEBOX_ICODE_SLI_S_BlockProtectionStatus { int LockBit; int ReadPasswordProtected; int WritePasswordProtected; int PageProtectionLock; } BLUEBOX_ICODE_SLI_S_BlockProtectionStatus;</pre>	

2.4.4 BLUEBOX_Registration

Name	BLUEBOX_Registration	
Description	Registration struct	
Data fields	TagType	Tag type
	Id	Pointer to the tag ID
	Length	The length of the tag ID in bytes
	Antenna	The antenna which identified the tag
	Input	The input which activated the identification procedure
Syntax	<pre>typedef struct BLUEBOX_Registration { BLUEBOX_TagType TagType; unsigned char* Id; int Length; BLUEBOX_Antenna Antenna; BLUEBOX_Input Input; } BLUEBOX_Registration;</pre>	

2.5 Functions

2.5.1 BLUEBOX_GetSwRelease

Name	BLUEBOX_GetSwRelease
Reader	All readers

Description	This function gets the software release of the library	
Parameters	[out] SwRel	Software release of the library
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetSwRelease(char* SwRel);</pre>	

2.5.2 BLUEBOX_Init

Name	BLUEBOX_Init	
Reader	All readers	
Description	This function creates an opaque handle to identify a module attached to PC	
Parameters	[out] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InitError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Init(BLUEBOX_Handle* Handle);</pre>	

2.5.3 BLUEBOX_End

Name	BLUEBOX_End	
Reader	All readers	
Description	This function notifies the library the end of operation and frees the allocated memory. Implicitly calls the BLUEBOX_Close function if the connection with the reader is open.	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_End(BLUEBOX_Handle* Handle);</pre>	

2.5.4 BLUEBOX_Open

Name	BLUEBOX_Open	
Reader	All readers	
Description	This function opens the connection with the reader. If already connected, it tries to close the connection and then opens it.	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_GenericError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Open(BLUEBOX_Handle* Handle);</pre>	

2.5.5 BLUEBOX_Close

Name	BLUEBOX_Close	
Reader	All readers	
Description	This function closes the connection with the reader.	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError	

Syntax	<code>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Close (BLUEBOX_Handle* Handle);</code>
---------------	--

2.5.6 BLUEBOX_SetAddress

Name	BLUEBOX_SetAddress	
Reader	All readers	
Description	This function sets the reader address.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Address	Address to use to communicate with the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle	
Syntax	<code>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SetAddress (BLUEBOX_Handle* Handle, unsigned char Address);</code>	

2.5.7 BLUEBOX_SetDevice

Name	BLUEBOX_SetDevice	
Reader	All readers	
Description	This function sets the reader type, frequency range and other parameters needed to communicate correctly with the reader.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Type	The reader type string. Use one of the strings listed below: <ul style="list-style-type: none"> • "DESKTOP": Desktop reader • "INDUSTRIAL": Industrial reader • "TINYOEM": OEM reader like OEM HF or OEM LF • "EASYOEM": OEM reader like OEM HF E • "BB OEM": OEM reader like OEM UHF • "PORTAL": Portal reader • "BB2 DESKTOP": Gen2 desktop reader • "BB2 INDUSTRIAL": Gen2 industrial reader • "BB2 BASIC": Gen2 basic reader • "CX": CX readers
	[in] Frequency	The reader frequency string. Use one of the strings listed below: <ul style="list-style-type: none"> • "LF": Low Frequency (125 kHz) • "HF": High Frequency (13.56 MHz) • "UHF": Ultra High Frequency (860 – 960 MHz) • "MICROWAVE": Microwave 2.4 GHz
	[in] Range	The reader range string. Use one of the strings listed below: <ul style="list-style-type: none"> • "SHORT": Short range • "MID": Mid range • "LONG": Long range
	[in] Antennas	The reader antennas string. Use one of the values listed below: <ul style="list-style-type: none"> • "SINGLE": Single antenna • "DUAL": Dual antennas • "QUAD": Quad antennas
	[in] Major	The firmware version major number
	[in] Minor	The firmware version minor number
	[in] Variant	The firmware variant. One char which identifies the firmware variant.

Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidParams
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SetDevice (BLUEBOX_Handle* Handle, char* Type, char* Frequency, char* Range, char* Antennas, int Major, int Minor, char Variant);</pre>

2.5.8 BLUEBOX_GetDevice

Name	BLUEBOX_GetDevice	
Reader	All readers	
Description	This function gets the reader type, frequency range and other parameters needed to communicate correctly with the reader.	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Type	The reader type string. Use one of the strings listed below: <ul style="list-style-type: none"> • "DESKTOP": Desktop reader • "INDUSTRIAL": Industrial reader • "TINYOEM": OEM reader like OEM HF or OEM LF • "EASYOEM": OEM reader like OEM HF E • "BB OEM": OEM reader like OEM UHF • "PORTAL": Portal reader • "BB2 DESKTOP": Gen2 desktop reader • "BB2 INDUSTRIAL": Gen2 industrial reader • "BB2 BASIC": Gen2 basic reader • "CX": CX readers
	[out] Frequency	The reader frequency string. Use one of the strings listed below: <ul style="list-style-type: none"> • "LF": Low Frequency (125 kHz) • "HF": High Frequency (13.56 MHz) • "UHF": Ultra High Frequency (860 – 960 MHz) • "MICROWAVE": Microwave 2.4 GHz
	[out] Range	The reader range string. Use one of the strings listed below: <ul style="list-style-type: none"> • "SHORT": Short range • "MID": Mid range • "LONG": Long range
	[out] Antennas	The reader antennas string. Use one of the values listed below: <ul style="list-style-type: none"> • "SINGLE": Single antenna • "DUAL": Dual antennas • "QUAD": Quad antennas
	[out] Major	The firmware version major number
	[out] Minor	The firmware version minor number
	[out] Variant	The firmware variant. One char which identifies the firmware variant.
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetDevice (BLUEBOX_Handle* Handle, char* Type, char* Frequency, char* Range, char* Antennas, int* Major, int* Minor, char* Variant);</pre>	

2.5.9 BLUEBOX_SetChannel

Name	BLUEBOX_SetChannel	
Reader	All readers	
Description	This function sets the notification channel to use with the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Channel	Channel to use string. Use one of the strings listed below: <ul style="list-style-type: none"> • "RS232": RS232 • "RS485": RS485 • "TCP": TCP
	[in] Settings	Channel settings string. Use one of the forms listed below and depending on the channel to use: <ul style="list-style-type: none"> • RS232: "<port name>,<baud rate>,<data bits>,<parity>,<stop bits>,<retx>,<timeout>" (e.g. "COM1,19200,8,n,,1,5,60000") • RS485: "<port name>,<baud rate>,<data bits>,<parity>,<stop bits>,<retx>,<timeout>" (e.g. "COM1,19200,8,n,,1,5,60000") • TCP: "<ip>:<port>,<timeout>" (e.g. "192.168.4.200:3000,60000")
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidParams	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SetChannel (BLUEBOX_Handle* Handle, char* Channel, char* Settings);</pre>	
Remarks	The <timeout> field is expressed in ms. RS232 is also used with USB Virtual Com interfaces.	

2.5.10 BLUEBOX_GetFwRelease

Name	BLUEBOX_GetFwRelease	
Reader	All readers	
Description	This function gets the firmware release of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Reader	The reader to read the version firmware. Use one of the values listed below and defined in BLUEBOX_Reader: <ul style="list-style-type: none"> • BLUEBOX_PRIMARY_READER: The primary reader • BLUEBOX_AUXILIARY_1_READER: The 1st auxiliary reader • BLUEBOX_AUXILIARY_2_READER: The 2nd auxiliary reader
	[out] FwRel	The firmware release of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidParams, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetFwRelease (BLUEBOX_Handle* Handle, BLUEBOX_READER Reader, char* FwRel);</pre>	
Remarks	The function must be called after opening the connection with the reader.	

2.5.11 BLUEBOX_Reset

Name	BLUEBOX_Reset
Reader	All readers

Description	This function resets the reader	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Reset(BLUEBOX_Handle* Handle);</pre>	

2.5.12 BLUEBOX_GetDateTime

Name	BLUEBOX_GetDateTime	
Reader	All BLUEBOX GEN2 INDUSTRIAL readers	
Description	This function gets the date/time set in the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[out] DateTime	The date/time string in BCD format (YYYYMMDDhhmmss). It is a null terminated string.
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetDateTime(BLUEBOX_Handle* Handle, char* DateTime);</pre>	

2.5.13 BLUEBOX_SetDateTime

Name	BLUEBOX_SetDateTime	
Reader	All BLUEBOX GEN2 INDUSTRIAL readers	
Description	This function sets the date/time set in the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] DateTime	The date/time string in BCD format (YYYYMMDDhhmmss). It is a null terminated string.
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SetDateTime(BLUEBOX_Handle* Handle, char* DateTime);</pre>	

2.5.14 BLUEBOX_ReadParameters

Name	BLUEBOX_ReadParameters	
Reader	All readers	
Description	This function reads the general parameters of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Parameters	General parameters set in the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadParameters(BLUEBOX_Handle* Handle,</pre>	

	<code>unsigned char* Parameters);</code>
Remarks	This function must be called after opening the connection with the reader and after reading the firmware version. See the reader technical manual for the Parameters format.

2.5.15 BLUEBOX_WriteParameters

Name	BLUEBOX_WriteParameters	
Reader	All readers	
Description	This function writes the general parameters of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Parameters	General parameters to be set in the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_InvalidParams	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteParameters (BLUEBOX_Handle* Handle, unsigned char* Parameters);</pre>	
Remarks	See the reader technical manual for the Parameters format.	

2.5.16 BLUEBOX_DefaultParameters

Name	BLUEBOX_DefaultParameters	
Reader	All readers	
Description	This function resets the parameters to the factory default values.	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_DefaultParameters (BLUEBOX_Handle* Handle);</pre>	

2.5.17 BLUEBOX_ReadSerialNumber

Name	BLUEBOX_ReadSerialNumber	
Reader	All BLUEBOX GEN2 INDUSTRIAL readers, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function reads the serial number of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Serial	Serial number set in the reader. 6-bytes long, in BCD format
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadSerialNumber (BLUEBOX_Handle* Handle, unsigned char* Serial);</pre>	

2.5.18 BLUEBOX_ReadMACAddress

Name	BLUEBOX_ReadMACAddress	
Reader	All BLUEBOX GEN2 INDUSTRIAL readers, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function reads the MAC address of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[out] MAC	MAC address of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadMACAddress (BLUEBOX_Handle* Handle, unsigned char* MAC);</pre>	

2.5.19 BLUEBOX_ReadConfiguration

Name	BLUEBOX_ReadConfiguration	
Reader	All readers	
Description	This function reads a configuration page of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Page	The configuration page number to read
	[out] Config	Configuration set in the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadConfiguration (BLUEBOX_Handle* Handle, int Page unsigned char* Config);</pre>	
Remarks	The configuration buffer size in bytes depends on the page and it is 7 bytes long for pages between 0x00 and 0x7F, and 14 bytes long for pages between 0x80 and 0xFF.	

2.5.20 BLUEBOX_WriteConfiguration

Name	BLUEBOX_WriteConfiguration	
Reader	All readers	
Description	This function writes a configuration page of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Page	The configuration page number to write
	[in] Config	Configuration to be set in the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_InvalidParams	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteConfiguration (BLUEBOX_Handle* Handle, int Page unsigned char* Config);</pre>	

Remarks	The configuration buffer size in bytes depends on the page and it is 7 bytes long for pages between 0x00 and 0x7F, and 14 bytes long for pages between 0x80 and 0xFF.
----------------	---

2.5.21 BLUEBOX_DefaultConfiguration

Name	BLUEBOX_DefaultConfiguration	
Reader	All readers	
Description	This function resets the configuration to the factory default values	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_DefaultConfiguration(BLUEBOX_Handle* Handle);</pre>	

2.5.22 BLUEBOX_DataRequest

Name	BLUEBOX_DataRequest	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function reads data from the buffer. The tags array contains the tag ID and other information related to every tag such as antenna and ID length.	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_MemoryError, BLUEBOX_TagNotFound	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_DataRequest(BLUEBOX_Handle* Handle, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	

2.5.23 BLUEBOX_QueueRequest

Name	BLUEBOX_QueueRequest	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function reads data from the queue. The tags array contains the tag ID and other information related to every tag such as antenna and ID length.	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_InvalidCommand, BLUEBOX_ConnectionError, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_MemoryError, BLUEBOX_TagNotFound	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_QueueRequest(BLUEBOX_Handle* Handle, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	

2.5.24 BLUEBOX_FreeTagsMemory

Name	BLUEBOX_FreeTagsMemory	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function frees the memory allocated to store tags by BLUEBOX_DataRequest or BLUEBOX_QueueRequest or inventory commands.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Tags	The array containing the tags read
	[in] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_FreeTagsMemory (BLUEBOX_Handle* Handle, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	

2.5.25 BLUEBOX_AllocateNotifyChannel

Name	BLUEBOX_AllocateNotifyChannel	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function allocates a notification channel to start tag notification.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Address	The address of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidParams, BLUEBOX_AllocationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_AllocateNotifyChannel (BLUEBOX_Handle* Handle, unsigned char Address);</pre>	

2.5.26 BLUEBOX_DeallocateNotifyChannel

Name	BLUEBOX_DeallocateNotifyChannel	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function deallocates a notification channel to stop tag notification.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Address	The address of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidChannel, BLUEBOX_AllocationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_DeallocateNotifyChannel (BLUEBOX_Handle* Handle, unsigned char Address);</pre>	

2.5.27 BLUEBOX_GetNotification

Name	BLUEBOX_GetNotification	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function reads data from the notification buffer. The tags array contains tag ID and other information related to every tag such as antenna and ID length.	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Tags	The array containing the tags read

	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_MemoryError, BLUEBOX_TagNotFound, BLUEBOX_AllocationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetNotification(BLUEBOX_Handle* Handle, BLUEBOX_Notify** Tags, int* TagsNo);</pre>	

2.5.28 BLUEBOX_FreeNotifyMemory

Name	BLUEBOX_FreeNotifyMemory	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function frees the memory allocated to store tags by the BLUEBOX_GetNotification command.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Tags	The array containing the tags read
	[in] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_FreeNotifyMemory(BLUEBOX_Handle* Handle, BLUEBOX_Notify** Tags, int TagsNo);</pre>	

2.5.29 BLUEBOX_SetOutput

Name	BLUEBOX_SetOutput	
Reader	BLUEBOX OEM, BLUEBOX INDUSTRIAL, BLUEBOX GEN2 OEM, INDUSTRIAL and BASIC readers, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function sets an output behaviour.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Output	The value that identifies the output. Use one of the values defined in BLUEBOX_Output
	[in] Period	Activation period from 1 (0x01) to 99 (0x63) seconds. Use 0x80 to continuously deactivate the output and 0x81 to continuously activate the output.
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_InvalidParams	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SetOutput(BLUEBOX_Handle* Handle, BLUEBOX_Output output, unsigned char Period);</pre>	

2.5.30 BLUEBOX_GetReaderStatus

Name	BLUEBOX_GetReaderStatus	
Reader	All readers	
Description	This function reads the status of the reader	

Parameters	[in] Handle	The handle that identifies the reader
	[out] Status	The status of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetReaderStatus(BLUEBOX_Handle* Handle, BLUEBOX_ReaderStatus* Status);</pre>	
Remarks	See the reader technical manual for the Status format.	

2.5.31 BLUEBOX_GetTemperature

Name	BLUEBOX_GetTemperature	
Reader	All BLUEBOX GEN2 INDUSTRIAL readers, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function reads the internal temperature of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Temperature	The internal temperature of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetTemperature(BLUEBOX_Handle* Handle, BLUEBOX_ReaderStatus* Status);</pre>	

2.5.32 BLUEBOX_RfOnOff

Name	BLUEBOX_RfOnOff	
Reader	All readers except of BLUEBOX PORTAL UHF	
Description	This function sets RF ON/OFF	
Parameters	[in] Handle	The handle that identifies the reader
	[in] OnOff	Flag to activate / deactivate the RF. Use one of the values listed below: <ul style="list-style-type: none"> 0: to deactivate the RF all other values: to activate the RF
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_RfOnOff(BLUEBOX_Handle* Handle, short OnOff);</pre>	

2.5.33 BLUEBOX_SelectiveRfOnOff

Name	BLUEBOX_SelectiveRfOnOff	
Reader	BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function sets selectively RF ON/OFF	

Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to switch On / Off
	[in] OnOff	Flag to activate / deactivate the RF. Use one of the values listed below: <ul style="list-style-type: none"> 0: to deactivate the RF all other values: to activate the RF
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SelectiveRFOnOff (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, short OnOff);</pre>	

2.5.34 BLUEBOX_ReadID_EM4305

Name	BLUEBOX_ReadID_EM4305	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read the ID of an EM4305 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's ID. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] TagType	Value as defined in BLUEBOX_TagType: BLUEBOX_EM4305
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadID_EM4305 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_TagType* TagType, void* Data);</pre>	

2.5.35 BLUEBOX_Write_EM4305

Name	BLUEBOX_Write_EM4305	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write an EM4305 tag with one of the codes defined below <ul style="list-style-type: none"> BLUEBOX SHORT: 5 bytes, UNIQUE equivalent BLUEBOX MEDIUM: 10 bytes BLUEBOX LONG: 20 bytes 	
Parameters	[in] Handle	The handle that identifies the reader

	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Data	The data to write in the tag's memory
	[in] Length	The number of bytes to write. Allowed values are 5, 10 and 20
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_EM4305 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Data int Length);</pre>	

2.5.36 BLUEBOX_ReadID_T5557

Name	BLUEBOX_ReadID_T5557	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read the ID of a T5557 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's ID. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] TagType	Value as defined in BLUEBOX_TagType: BLUEBOX_T5557
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadID_T5557 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_TagType* TagType, void* Data);</pre>	

2.5.37 BLUEBOX_Write_T5557

Name	BLUEBOX_Write_T5557	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a T5557 tag with one of the codes defined below <ul style="list-style-type: none"> BLUEBOX SHORT: 5 bytes, UNIQUE equivalent BLUEBOX MEDIUM: 10 bytes BLUEBOX LONG: 20 bytes 	

Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Data	The data to write in the tag's memory
	[in] Length	The number of bytes to write. Allowed values are 5, 10 and 20
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_T5557(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Data int Length);</pre>	

2.5.38 BLUEBOX_ReadID_Q5

Name	BLUEBOX_ReadID_Q5	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read the ID of a Q5 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's ID. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] TagType	Value as defined in BLUEBOX_TagType: BLUEBOX_Q5
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadID_Q5(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_TagType* TagType, void* Data);</pre>	

2.5.39 BLUEBOX_Write_Q5

Name	BLUEBOX_Write_Q5	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a Q5 tag with one of the codes defined below <ul style="list-style-type: none"> BLUEBOX SHORT: 5 bytes, UNIQUE equivalent BLUEBOX MEDIUM: 10 bytes 	

	<ul style="list-style-type: none"> BLUEBOX LONG: 20 bytes 	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Data	The data to write in the tag's memory
	[in] Length	The number of bytes to write. Allowed values are 5, 10 and 20
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_Q5 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Data int Length);</pre>	

2.5.40 BLUEBOX_ReadID_HITAG1

Name	BLUEBOX_ReadID_HITAG1	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read the ID of a HITAG 1 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's ID. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] TagType	Value as defined in BLUEBOX_TagType: BLUEBOX_HITAG_1
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadID_HITAG1 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_TagType* TagType, void* Data);</pre>	

2.5.41 BLUEBOX_ReadPage_HITAG1

Name	BLUEBOX_ReadPage_HITAG1	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a page of a HITAG 1 tag	
Parameters	[in] Handle	The handle that identifies the reader

	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Page	The page of the tag's memory to read (16 – 63)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadPage_HITAG1 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, void* Data);</pre>	

2.5.42 BLUEBOX_WritePage_HITAG1

Name	BLUEBOX_WritePage_HITAG1	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a page of a HITAG 1 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Page	The page of the tag's memory to write (16 – 63)
	[in] Data	The data written to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WritePage_HITAG1 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, void* Data);</pre>	

2.5.43 BLUEBOX_ReadID_HITAGS

Name	BLUEBOX_ReadID_HITAGS	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read the ID of a HITAG S tag	

Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's ID. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] TagType	Values as defined in BLUEBOX_TagType: BLUEBOX_HITAG_S256, BLUEBOX_HITAG_S2048
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadID_HITAGS (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_TagType* TagType, void* Data);</pre>	

2.5.44 BLUEBOX_Write_HITAGS

Name	BLUEBOX_Write_HITAGS	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a HITAG S tag with one of the codes defined below <ul style="list-style-type: none"> BLUEBOX SHORT: 5 bytes, UNIQUE equivalent BLUEBOX MEDIUM: 10 bytes 	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Data	The data to write in the tag's memory
	[in] Length	The number of bytes to write. Allowed values are 5 and 10
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_HITAGS (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Data int Length);</pre>	

2.5.45 BLUEBOX_ReadPage_HITAGS

Name	BLUEBOX_ReadPage_HITAGS	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	

Description	This function allows to read a page of a HITAG S tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Page	The page of the tag's memory to read (0 – 63)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadPage_HITAGS (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, void* Data);</pre>	

2.5.46 BLUEBOX_WritePage_HITAGS

Name	BLUEBOX_WritePage_HITAGS	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a page of a HITAG S tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Page	The page of the tag's memory to write (0 – 63)
	[in] Data	The data written to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WritePage_HITAGS (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, void* Data);</pre>	

2.5.47 BLUEBOX_ReadID_TITAN

Name	BLUEBOX_ReadID_TITAN
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP

	LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read the ID of a TITAN tag (ID + SN)	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's ID. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] TagType	Value as defined in BLUEBOX_TagType: BLUEBOX_TITAN
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadID_TITAN(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_TagType* TagType, void* Data);</pre>	

2.5.48 BLUEBOX_Reset_TITAN

Name	BLUEBOX_Reset_TITAN	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to reset a TITAN tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to reset the tag. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Reset_TITAN(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna);</pre>	

2.5.49 BLUEBOX_Login_TITAN

Name	BLUEBOX_Login_TITAN	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to login a TITAN tag	
Parameters	[in] Handle	The handle that identifies the reader

	[in] Antenna	The antenna to use to login the tag. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Password	The password to use to login the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Login_TITAN(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Password);</pre>	

2.5.50 BLUEBOX_WritePassword_TITAN

Name	BLUEBOX_WritePassword_TITAN	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write the password of a TITAN tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to login the tag. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] OldPwd	The old password of the tag
	[in] NewPwd	The new password of the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WritePassword_TITAN(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* OldPwd, void* NewPwd);</pre>	

2.5.51 BLUEBOX_SelectiveRead_TITAN

Name	BLUEBOX_SelectiveRead_TITAN	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to selectively read of a TITAN tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2

	[in] Address	The address of the memory to read
	[in] Length	The number of words (4 bytes long) to read from the tag's memory
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SelectiveRead_TITAN(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, int Address, int Length, void* Data);</pre>	

2.5.52 BLUEBOX_SelectiveWrite_TITAN

Name	BLUEBOX_SelectiveWrite_TITAN	
Reader	BLUEBOX DESKTOP LF, BLUEBOX OEM LF, BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to selectively write to a TITAN tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Address	The address of the memory to write
	[in] Length	The number of words (4 bytes long) to write to the tag's memory
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SelectiveWrite_TITAN(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, int Address, int Length, void* Data);</pre>	

2.5.53 BLUEBOX_Inventory_ISO15693

Name	BLUEBOX_Inventory_ISO15693	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL.	
Description	This function sends an inventory command with anticollision to read all the ISO 15693 tags. The tags array contains all the tag IDs and other information related to the tags such as antenna and ID length	

Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_NOANT: No antenna info, use only with quad channel readers BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] UseAFI	Flag to use the AFI during inventory. Accepted values: 0: No AFI; else uses AFI
	[in] AFI	The AFI field to use during inventory. Optional parameter needed when UseAFI != 0
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Inventory_ISO15693(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, short UseAFI, unsigned char AFI, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	

2.5.54 BLUEBOX_ReadPage_ISO15693

Name	BLUEBOX_ReadPage_ISO15693	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a page of an ISO 15693 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_NOANT: No antenna info, use only with quad channel readers BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Page	The page of the tag's memory to read (0 – 255)
	[in] Size	The size of the page to read in bytes (4, 8)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	

Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadPage_ISO15693 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, int Size, void* Data);</pre>
---------------	---

2.5.55 BLUEBOX_ReadMultiPage_ISO15693

Name	BLUEBOX_ReadMultiPage_ISO15693	
Reader	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL	
Description	This function allows to read multiple pages of an ISO 15693 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to read the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_NOANT: No antenna info, use only with quad channel readers • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Page	The page of the tag's memory to read (0 – 255)
	[in] Size	The size of the page to read in bytes (4, 8)
	[in] Count	The number of pages to read
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadMultiPage_ISO15693 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, int Size, int Count, void* Data);</pre>	

2.5.56 BLUEBOX_WritePage_ISO15693

Name	BLUEBOX_WritePage_ISO15693	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a page of an ISO 15693 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_NOANT: No antenna info, use only with quad channel readers • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write

	[in] Page	The page of the tag's memory to write (0 – 255)
	[in] Size	The size of the page to write in bytes (4, 8)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WritePage_ISO15693(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, int Size, void* Data);</pre>	

2.5.57 BLUEBOX_WriteMultiPage_ISO15693

Name	BLUEBOX_WriteMultiPage_ISO15693	
Reader	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL	
Description	This function allows to write multiple pages of an ISO 15693 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_NOANT: No antenna info, use only with quad channel readers • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Page	The page of the tag's memory to write (0 – 255)
	[in] Size	The size of the page to write in bytes (4, 8)
	[in] Count	The number of pages to write
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteMultiPage_ISO15693(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page, int Size, int Count, void* Data);</pre>	

2.5.58 BLUEBOX_LockPage_ISO15693

Name	BLUEBOX_LockPage_ISO15693	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to lock a page of an ISO 15693 tag	

Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to lock the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_NOANT: No antenna info, use only with quad channel readers • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to lock
	[in] Page	The page of the tag's memory to lock (0 – 255)
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_LockPage_ISO15693 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Page);</pre>	

2.5.59 BLUEBOX_Write_AFI_ISO15693

Name	BLUEBOX_Write_AFI_ISO15693	
Reader	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL	
Description	This function allows to write the AFI of an ISO 15693 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag to write
	[in] Afi	The tag's AFI to be written
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_AFI_ISO15693 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, unsigned char Afi);</pre>	

2.5.60 BLUEBOX_Lock_AFI_ISO15693

Name	BLUEBOX_Lock_AFI_ISO15693	
Reader	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL	
Description	This function allows to lock the AFI of an ISO 15693 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to lock the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1

	[in] Id	The ID of the tag to lock
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Lock_AFI_ISO15693(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id);</pre>	

2.5.61 BLUEBOX_GetRandomNumber_ICODE_SLI_S

Name	BLUEBOX_GetRandomNumber_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to get a random number from an ICODE SLI-S tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use to write the tag's memory. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[out] Random	The random number received from the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetRandomNumber_ICODE_SLI_S(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, void* Random);</pre>	

2.5.62 BLUEBOX_SetPassword_ICODE_SLI_S

Name	BLUEBOX_SetPassword_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to transmit a password to an ICODE SLI-S tag to get access to different protected functionalities of the tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] PwdId	The password identifier which identifies the type of the password to transmit
	[in] Password	The password to transmit to the tag
	[in] Random	The random number previously received from the tag

Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_SetPassword_ICODE_SLI_S(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, BLUEBOX_ICODE_SLI_S_PasswordIdentifier PwdId, void* Password, void* Random);</pre>

2.5.63 BLUEBOX_WritePassword_ICODE_SLI_S

Name	BLUEBOX_WritePassword_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a new password to an ICODE SLI-S tag if the related old password has already been transmitted with a Set Password command before and the addressed password is not locked.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] PwdId	The password identifier which identifies the type of the password to write
	[in] Password	The new password to write to the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WritePassword_ICODE_SLI_S(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, BLUEBOX_ICODE_SLI_S_PasswordIdentifier PwdId, void* Password);</pre>	

2.5.64 BLUEBOX_LockPassword_ICODE_SLI_S

Name	BLUEBOX_LockPassword_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to lock a password of an ICODE SLI-S tag if the related old password has already been transmitted with a Set Password command before and the addressed password is not locked.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna.

		<ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] PwdId	The password identifier which identifies the type of the password to lock
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_LockPassword_ICODE_SLI_S (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, BLUEBOX_ICODE_SLI_S PasswordIdentifier PwdId);</pre>	

2.5.65 BLUEBOX_64BitPasswordProtection_ICODE_SLI_S

Name	BLUEBOX_64BitPasswordProtection_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to activate the 64-bit password protection. This mode can be enabled if both the Read and Write password have already been transmitted with a Set Password command before.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] PwdId	The password identifier which identifies the type of the password to write
	[in] Password	The new password to write to the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_64BitPasswordProtection_ICODE_SLI_S (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id);</pre>	

2.5.66 BLUEBOX_ProtectPage_ICODE_SLI_S

Name	BLUEBOX_ProtectPage_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to change the protection condition of a page of an ICODE SLI-S tag if the related passwords have already been transmitted with a Set Password command before and the addressed page is not locked	
Parameters	[in] Handle	The handle that identifies the reader

	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] PageNo	The number of the page
	[in] Status	The protection status. Use one of the values defined in the BLUEBOX_ICODE_SLI_S_ProtectionStatus enum
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ProtectPage_ICODE_SLI_S (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int PageNo, BLUEBOX_ICODE_SLI_S_ProtectionStatus Status);</pre>	

2.5.67 BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S

Name	BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to lock the protection condition of a page of an ICODE SLI-S tag if the related passwords have already been transmitted with a Set Password command before	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] PageNo	The number of the page
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int PageNo);</pre>	

2.5.68 BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S

Name	BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to get the block protection status of the requested blocks of an ICODE SLI-S tag.	
Parameters	[in] Handle	The handle that identifies the reader

	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
	[in] BlockNo	The number of the first block
	[in] Length	The number of blocks
	[out] Status	The block protection status of the requested blocks array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int BlockNo, int Length, BLUEBOX_ICODE_SLI_S ProtectionStatus* Status);</pre>	

2.5.69 BLUEBOX_Destroy_SLI_S_ICODE_SLI_S

Name	BLUEBOX_Destroy_SLI_S_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to destroy an ICODE SLI-S tag. It can be destroyed if the Destroy SLI-S password has been transmitted before. This command is irreversible.	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Destroy_SLI_S_ICODE_SLI_S (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id);</pre>	

2.5.70 BLUEBOX_EnablePrivacy_ICODE_SLI_S

Name	BLUEBOX_EnablePrivacy_ICODE_SLI_S	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to set an ICODE SLI-S tag into Privacy mode if the Privacy password has already been transmitted before	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna.

		<ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1
	[in] Id	The ID of the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_EnablePrivacy_ICODE_SLI_S(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id);</pre>	

2.5.71 BLUEBOX_Inventory_ISO14443A

Name	BLUEBOX_Inventory_ISO14443A	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function sends an inventory command with anticollision to read all ISO 14443A tags. The tags array contains all the tag IDs and other information related to every tag, such as antenna and ID length	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Inventory_ISO14443A(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	
Remarks	Reads only one ISO 14443A tag	

2.5.72 BLUEBOX_ReadBlock_MIFARE_1k

Name	BLUEBOX_ReadBlock_MIFARE_1k	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of a MIFARE 1k tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1

		<ul style="list-style-type: none"> • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] KeyType	The key type to use. Use one of the values listed below and defined in BLUEBOX_MifareKey: <ul style="list-style-type: none"> • BLUEBOX_MIFARE_KEY_A: Key A • BLUEBOX_MIFARE_KEY_B: Key B
	[in] Key	The key to use to read the tag's memory
	[in] Block	The block of the tag's memory to read (0 – 63)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_MIFARE_1k(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, BLUEBOX_MifareKey KeyType, void* Key, int Block, void* Data);</pre>	

2.5.73 BLUEBOX_WriteBlock_MIFARE_1k

Name	BLUEBOX_WriteBlock_MIFARE_1k	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of a MIFARE 1k tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] KeyType	The key type to use. Use one of the values listed below and defined in BLUEBOX_MifareKey: <ul style="list-style-type: none"> • BLUEBOX_MIFARE_KEY_A: Key A • BLUEBOX_MIFARE_KEY_B: Key B
	[in] Key	The key to use to write the tag's memory
	[in] Block	The block of the tag's memory to write (0 – 63)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_MIFARE_1k(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id,</pre>	

```
BLUEBOX_MifareKey KeyType, void* Key,
int Block, void* Data);
```

2.5.74 BLUEBOX_ReadBlock_MIFARE_4k

Name	BLUEBOX_ReadBlock_MIFARE_4k	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of a MIFARE 4k tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] KeyType	The key type to use. Use one of the values listed below and defined in BLUEBOX_MifareKey: <ul style="list-style-type: none"> BLUEBOX_MIFARE_KEY_A: Key A BLUEBOX_MIFARE_KEY_B: Key B
	[in] Key	The key to use to read the tag's memory
	[in] Block	The block of the tag's memory to read (0 – 255)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_MIFARE_4k(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, BLUEBOX_MifareKey KeyType, void* Key, int Block, void* Data);</pre>	

2.5.75 BLUEBOX_WriteBlock_MIFARE_4k

Name	BLUEBOX_WriteBlock_MIFARE_4k	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of a MIFARE 4k tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write

	[in] KeyType	The key type to use. Use one of the values listed below and defined in BLUEBOX_MifareKey: <ul style="list-style-type: none"> BLUEBOX_MIFARE_KEY_A: Key A BLUEBOX_MIFARE_KEY_B: Key B
	[in] Key	The key to use to write the tag's memory
	[in] Block	The block of the tag's memory to write (0 – 255)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_MIFARE_4k(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, BLUEBOX_MifareKey KeyType, void* Key, int Block, void* Data);</pre>	

2.5.76 BLUEBOX_ReadBlock_MIFARE_Ultralight

Name	BLUEBOX_ReadBlock_MIFARE_Ultralight	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of a MIFARE Ultralight tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Block	The block of the tag's memory to read (0 – 15)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_MIFARE_Ultralight(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.77 BLUEBOX_WriteBlock_MIFARE_Ultralight

Name	BLUEBOX_WriteBlock_MIFARE_Ultralight	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX	

	GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of a MIFARE Ultralight tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Block	The block of the tag's memory to write (0 – 15)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_MIFARE_Ultralight (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.78 BLUEBOX_ReadBlock_NTAG213

Name	BLUEBOX_ReadBlock_NTAG213	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of an NTAG213 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Block	The block of the tag's memory to read (0 – 44)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_NTAG213 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.79 BLUEBOX_WriteBlock_NTAG213

Name	BLUEBOX_WriteBlock_NTAG213
-------------	----------------------------

Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of an NTAG213 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Block	The block of the tag's memory to write (0 – 44)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_NTAG213(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.80 BLUEBOX_ReadBlock_NTAG215

Name	BLUEBOX_ReadBlock_NTAG215	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of an NTAG215 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Block	The block of the tag's memory to read (0 – 134)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_NTAG215(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.81 BLUEBOX_WriteBlock_NTAG215

Name	BLUEBOX_WriteBlock_NTAG215	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of an NTAG215 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Block	The block of the tag's memory to write (0 – 134)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_NTAG215 (BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.82 BLUEBOX_ReadBlock_NTAG216

Name	BLUEBOX_ReadBlock_NTAG216	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of an NTAG216 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Block	The block of the tag's memory to read (0 – 230)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_NTAG216 (BLUEBOX_Handle* Handle,</pre>	


```
BLUEBOX_Antenna Antenna, void* Id,
int Block, void* Data);
```

2.5.83 BLUEBOX_WriteBlock_NTAG216

Name	BLUEBOX_WriteBlock_NTAG216	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of an NTAG216 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Block	The block of the tag's memory to write (0 – 230)
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_NTAG216(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.84 BLUEBOX_Inventory_ISO14443B

Name	BLUEBOX_Inventory_ISO14443B	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function sends an inventory command with anticollision to read all ISO 14443B tags. The tags array contains all the tag IDs and other information related to every tag, such as antenna and ID length	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> • BLUEBOX_ANT_1: Antenna 1 • BLUEBOX_ANT_2: Antenna 2
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle,	

	BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Inventory_ISO1443B(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_Tag** Tags, int* TagsNo);</pre>
Remarks	Reads only one ISO 1443B tag

2.5.85 BLUEBOX_ReadBlock_SR176

Name	BLUEBOX_ReadBlock_SR176	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to read a block of memory of a SR176 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to read
	[in] Block	The block of the tag's memory to read (0 – 63)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadBlock_SR176(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.86 BLUEBOX_WriteBlock_SR176

Name	BLUEBOX_WriteBlock_SR176	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function allows to write a block of memory of a SR176 tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[in] Id	The ID of the tag to write
	[in] Block	The block of the tag's memory to write (0 – 63)

	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteBlock_SR176(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, void* Id, int Block, void* Data);</pre>	

2.5.87 BLUEBOX_Inventory_PICOPASS

Name	BLUEBOX_Inventory_PICOPASS	
Reader	BLUEBOX DESKTOP HF, BLUEBOX OEM HF, BLUEBOX OEM HF E, BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	
Description	This function sends an inventory command with anticollision to read all Picopass tags. The tags array contains all the tag IDs and other information related to every tag, such as antenna and ID length	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Antenna	The antenna to use. Use one of the values listed below and defined in BLUEBOX_Antenna. <ul style="list-style-type: none"> BLUEBOX_ANT_1: Antenna 1 BLUEBOX_ANT_2: Antenna 2
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Inventory_PICOPASS(BLUEBOX_Handle* Handle, BLUEBOX_Antenna Antenna, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	

2.5.88 BLUEBOX_ReadRfParameters

Name	BLUEBOX_Inventory_ReadRfParameters	
Reader	BLUEBOX INDUSTRIAL HF MID/LONG RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE QUAND CHANNEL, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL ACTIVE, BLUEBOX PORTAL UHF, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 UHF MID RANGE SINGLE CHANNEL	
Description	This function reads the RF parameters of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Parameters	RF parameters of the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle,	

	BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadRfParameters (BLUEBOX_Handle* Handle, unsigned char* Parameters);</pre>
Remarks	See the reader technical manual for the Parameters format. This function could be replaced with BLUEBOX_ReadConfiguration (2.5.19)

2.5.89 BLUEBOX_WriteRfParameters

Name	BLUEBOX_Inventory_WriteRfParameters	
Reader	BLUEBOX INDUSTRIAL HF MID/LONG RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL HF LONG RANGE QUAND CHANNEL, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL ACTIVE, BLUEBOX PORTAL UHF, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 UHF MID RANGE SINGLE CHANNEL	
Description	This function writes the RF parameters of the reader	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Parameters	RF parameters to be set in the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_InvalidParams	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_WriteRfParameters (BLUEBOX_Handle* Handle, unsigned char* Parameters);</pre>	
Remarks	See the reader technical manual for the Parameters format. This function could be replaced with BLUEBOX_WriteConfiguration (2.5.202.5.19)	

2.5.90 BLUEBOX_Inventory_ISO18K6B

Name	BLUEBOX_Inventory_ISO18K6B	
Reader	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL	
Description	This function sends an inventory command with anticollision to read all ISO 18000-6B tags	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Inventory_ISO18K6B (BLUEBOX_Handle* Handle, BLUEBOX_Tag** Tags, int* TagsNo);</pre>	

2.5.91 BLUEBOX_Read_ISO18K6B

Name	BLUEBOX_Read_ISO18K6B
Reader	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL
Description	This function allows to read the memory of a ISO 18000-6B tag

Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to read
	[in] Address	The starting address of the tag's memory to read
	[in] Nblocks	The number of 8-bytes blocks to read (1..8)
	[out] Data	The data read from the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Read_ISO18K6B(BLUEBOX_Handle* Handle, void* Uid, int Address, int Nblocks, void* Data);</pre>	

2.5.92 BLUEBOX_Write_ISO18K6B

Name	BLUEBOX_Write_ISO18K6B	
Reader	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL	
Description	This function allows to write the memory of a ISO 18000-6B tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to write
	[in] Address	The starting address of the tag's memory to write
	[in] Length	The number of bytes to write (1..32)
	[out] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_ISO18K6B(BLUEBOX_Handle* Handle, void* Uid, int Address, int Length, void* Data);</pre>	

2.5.93 BLUEBOX_Inventory_ISO18K6C

Name	BLUEBOX_Inventory_ISO18K6C	
Reader	BLUEBOX OEM UHF, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 DESKTOP UHF, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function sends an inventory command with anticollision to read all ISO 18000-6C tags	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Tags	The array containing the tags read
	[out] TagsNo	The number of tags in the array
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle,	

	BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Inventory_ISO18K6C (BLUEBOX_Handle* Handle, BLUEBOX_Tag** Tags, int* TagsNo);</pre>

2.5.94 BLUEBOX_ProgramEPC_ISO18K6C

Name	BLUEBOX_ProgramEPC_ISO18K6C	
Reader	BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function allows to write the EPC of an ISO 18000-6C tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to be written
	[in] Pwd	The access password to write the tag. Set to 0 if no password is required
	[in] Length	The number of 2-bit words to write
	[in] Data	The data to be written into the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ProgramEPC_ISO18K6C (BLUEBOX_Handle* Handle, void* Uid, void* Pwd, int Length, void* Data);</pre>	

2.5.95 BLUEBOX_Read_ISO18K6C

Name	BLUEBOX_Read_ISO18K6C	
Reader	BLUEBOX OEM UHF, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 DESKTOP UHF, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function allows to read the memory of an ISO 18000-6C tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to read
	[in] Pwd	The access password to read the tag. Set to 0 if no password is required
	[in] Bank	The memory bank to read. One of the values listed below: <ul style="list-style-type: none"> BLUEBOX_ISO18K6C_BANK_RESERVED BLUEBOX_ISO18K6C_BANK_EPC BLUEBOX_ISO18K6C_BANK_TID BLUEBOX_ISO18K6C_BANK_USER
	[in] Address	The starting address of the tag's memory to read
	[in] Length	The number of 16-bit words to read
	[out] Data	The data read from the tag's memory

Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Read_ISO18K6C(BLUEBOX_Handle* Handle, void* Uid, void* Pwd, BLUEBOX_ISO18K6C_Bank Bank, int Address, int Length, void* Data);</pre>

2.5.96 BLUEBOX_Write_ISO18K6C

Name	BLUEBOX_Write_ISO18K6C	
Reader	BLUEBOX OEM UHF, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 DESKTOP UHF, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function allows to write the memory of an ISO 18000-6C tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to write
	[in] Pwd	The access password to write the tag. Set to 0 if no password is required
	[in] Bank	The memory bank to read. One of the values listed below: <ul style="list-style-type: none"> • BLUEBOX_ISO18K6C_BANK_RESERVED • BLUEBOX_ISO18K6C_BANK_EPC • BLUEBOX_ISO18K6C_BANK_TID • BLUEBOX_ISO18K6C_BANK_USER
	[in] Address	The starting address of the tag's memory to write
	[in] Length	The number of 2-bit words to write
	[in] Data	The data to write to the tag's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_ISO18K6C(BLUEBOX_Handle* Handle, void* Uid, void* Pwd, BLUEBOX_ISO18K6C_Bank Bank, int Address, int Length, void* Data);</pre>	

2.5.97 BLUEBOX_BlockWrite_ISO18K6C

Name	BLUEBOX_BlockWrite_ISO18K6C	
Reader	BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function allows to write the memory of an ISO 18000-6C tag using the BlockWrite command of the EPC Class1 Gen2 standard	
Parameters	[in] Handle	The handle that identifies the reader

	[in] Uid	The UID of the tag to write
	[in] Pwd	The access password to write the tag. Set to 0 if no password is required
	[in] Bank	The memory bank to read. One of the values listed below: <ul style="list-style-type: none"> BLUEBOX_ISO18K6C_BANK_RESERVED BLUEBOX_ISO18K6C_BANK_EPC BLUEBOX_ISO18K6C_BANK_TID BLUEBOX_ISO18K6C_BANK_USER
	[in] Address	The starting address of the tag's memory to write
	[in] Length	The number of 2-bit words to write
	[in] Data	The data to write to the tag's memory
	Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_BlockWrite_ISO18K6C(BLUEBOX_Handle* Handle, void* Uid, void* Pwd, BLUEBOX_ISO18K6C_Bank Bank, int Address, int Length, void* Data);</pre>	

2.5.98 BLUEBOX_Lock_ISO18K6C

Name	BLUEBOX_Write_ISO18K6C	
Reader	BLUEBOX OEM UHF, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 DESKTOP UHF, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function allows to lock the password and memory of an ISO 18000-6C tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to write
	[in] Pwd	The access password to write the tag. Must be != 0
	[in] KillPwd	To lock the kill password. One of the values listed below: <ul style="list-style-type: none"> BLUEBOX_ISO18K6C_TAG_PWD_PERM_ACCESSIBLE: Accessible from both opened and secured states BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_ACCESSIBLE: Permanently accessible from both opened and secured states and may never be locked BLUEBOX_ISO18K6C_TAG_PWD_PERM_SECURED_ACCESSIBLE: Accessible only from secured state BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_NOT_ACCESSIBLE: Not accessible from either opened or secured states BLUEBOX_ISO18K6C_TAG_PWD_PERM_NO_CHANGE: No change
	[in] AccessPwd	To lock the access password. One of the values listed below: <ul style="list-style-type: none"> BLUEBOX_ISO18K6C_TAG_PWD_PERM_ACCESSIBLE: Accessible from both opened and secured states

		<ul style="list-style-type: none"> • BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_ACCESSIBLE: Permanently accessible from both opened and secured states and may never be locked • BLUEBOX_ISO18K6C_TAG_PWD_PERM_SECURED_ACCESSIBLE: Accessible only from secured state • BLUEBOX_ISO18K6C_TAG_PWD_PERM_ALWAYS_NOT_ACCESSIBLE: Not accessible from either opened or secured states • BLUEBOX_ISO18K6C_TAG_PWD_PERM_NO_CHANGE: No change
	[in] EPCMemory	<p>To lock the EPC memory. One of the values listed below:</p> <ul style="list-style-type: none"> • BLUEBOX_ISO18K6C_TAG_MEM_PERM_WRITABLE: Writable from both opened and secured states • BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_WRITABLE: Permanently writable from both opened and secured states and may never be locked • BLUEBOX_ISO18K6C_TAG_MEM_PERM_SECURED_WRITABLE: Writable only from secured state • BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_NOT_WRITABLE: Not writable from either opened or secured state • BLUEBOX_ISO18K6C_TAG_MEM_PERM_NO_CHANGE: No change
	[in] TIDMemory	<p>To lock the TID memory. One of the values listed below:</p> <ul style="list-style-type: none"> • BLUEBOX_ISO18K6C_TAG_MEM_PERM_WRITABLE: Writable from both opened and secured states • BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_WRITABLE: Permanently writable from both opened and secured states and may never be locked • BLUEBOX_ISO18K6C_TAG_MEM_PERM_SECURED_WRITABLE: Writable only from secured state • BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_NOT_WRITABLE: Not writable from either opened or secured state • BLUEBOX_ISO18K6C_TAG_MEM_PERM_NO_CHANGE: No change
	[in] UserMemory	<p>To lock the User memory. One of the values listed below:</p> <ul style="list-style-type: none"> • BLUEBOX_ISO18K6C_TAG_MEM_PERM_WRITABLE: Writable from both opened and secured states • BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_WRITABLE: Permanently writable from both opened and secured states and may never be locked • BLUEBOX_ISO18K6C_TAG_MEM_PERM_SECURED_WRITABLE: Writable only from secured state • BLUEBOX_ISO18K6C_TAG_MEM_PERM_ALWAYS_NOT_WRITABLE: Not writable from either opened or secured state • BLUEBOX_ISO18K6C_TAG_MEM_PERM_NO_CHANGE: No change
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Write_ISO18K6C(BLUEBOX_Handle* Handle, void* Uid, void* Pwd, BLUEBOX_ISO18K6C_PasswordPermission KillPwd, BLUEBOX_ISO18K6C_PasswordPermission AccessPwd,</pre>	

```
BLUEBOX_ISO18K6C_MemoryPermission EPCMemory,
BLUEBOX_ISO18K6C_MemoryPermission TIDMemory,
BLUEBOX_ISO18K6C_MemoryPermission UserMemory);
```

2.5.99 Bluebox_Kill_ISO18K6C

Name	BLUEBOX_Kill_ISO18K6C	
Reader	BLUEBOX OEM UHF, BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL, BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL, BLUEBOX GEN2 OEM UHF, BLUEBOX GEN2 DESKTOP UHF, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF MID RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL, BLUEBOX CX UHF LONG RANGE DUAL CHANNEL	
Description	This function allows to kill an ISO 18000-6C tag	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Uid	The UID of the tag to kill
	[in] Pwd	The kill password to kill the tag
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_TagNotFound, BLUEBOX_TagError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_Kill_ISO18K6C (BLUEBOX_Handle* Handle, void* Uid, void* Pwd);</pre>	

2.5.100 BLUEBOX_FwUpgrade

Name	BLUEBOX_FwUpgrade	
Reader	All readers	
Description	This function allows to upgrade the readers firmware	
Parameters	[in] Handle	The handle that identifies the reader
	[in] FileName	The binary file name with the firmware to send to the reader
	[in] Reader	The reader to upgrade. Use one of the values listed below <ul style="list-style-type: none"> • BLUEBOX_PRIMARY_READER: Upgrade the primary reader • BLUEBOX_AUXILIARY_1_READER: Upgrade the 1st auxiliary reader • BLUEBOX_AUXILIARY_2_READER: Upgrade the 2nd auxiliary reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidParams, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError, BLUEBOX_GenericError, BLUEBOX_FileError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_FwUpgrade (BLUEBOX_Handle* Handle, BLUEBOX_UpgReader Reader, char* FileName);</pre>	

2.5.101 BLUEBOX_ReadNumberOfRegistrations

Name	BLUEBOX_ReadNumberOfRegistrations
-------------	-----------------------------------

Reader	BLUEBOX PORTAL UHF	
Description	This function reads the number of registrations saved in the reader's memory	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Registrations	The number of registrations saved in the reader's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadNumberOfRegistrations(BLUEBOX_Handle* Handle, int* Registrations);</pre>	

2.5.102 BLUEBOX_ReadOlderRegistration

Name	BLUEBOX_ReadOlderRegistration	
Reader	BLUEBOX PORTAL UHF	
Description	This function reads the older registration saved in the reader's memory	
Parameters	[in] Handle	The handle that identifies the reader
	[out] Index	The index of the registration read from the reader's memory
	[out] Registration	The registration read from the reader's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadOlderRegistration(BLUEBOX_Handle* Handle, int* Index, BLUEBOX_Registration* Registration);</pre>	

2.5.103 BLUEBOX_CancelOlderRegistration

Name	BLUEBOX_CancelOlderRegistration	
Reader	BLUEBOX PORTAL UHF	
Description	This function cancels the older registration saved in the reader's memory	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Index	The index of the registration to cancel from the reader's memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_CancelOlderRegistration(BLUEBOX_Handle* Handle, int Index);</pre>	

2.5.104 BLUEBOX_CancelAllRegistrations

Name	BLUEBOX_CancelAllRegistrations	
Reader	BLUEBOX PORTAL UHF	
Description	This function cancels all the registrations saved in the reader's memory	
Parameters	[in] Handle	The handle that identifies the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle,	

	BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_CancelAllRegistrations (BLUEBOX_Handle* Handle);</pre>

2.5.105 BLUEBOX_ReadPreviousRegistration

Name	BLUEBOX_ReadPreviousRegistration	
Reader	BLUEBOX PORTAL UHF	
Description	This function reads a previous registration saved in the reader’s memory	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Index	The index of the registration to read from the reader’s memory
	[out] Registration	The registrations read from the reader’s memory
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_ReadPreviousRegistration (BLUEBOX_Handle* Handle, int Index, BLUEBOX_Registration* Registration);	

2.5.106 BLUEBOX_GenericCommand

Name	BLUEBOX_GenericCommand	
Reader	All readers	
Description	This function sends a generic command using the BLUEBOX protocol	
Parameters	[in] Handle	The handle that identifies the reader
	[in] Command	The to send to the reader
	[out] Reply	The reply from the reader
Return	An error code about the execution of the function. One of the values listed below and defined in BLUEBOX_ErrorCodes: BLUEBOX_StatusOk, BLUEBOX_InvalidHandle, BLUEBOX_ConnectionError, BLUEBOX_InvalidCommand, BLUEBOX_TimeoutError, BLUEBOX_CommunicationError	
Syntax	<pre>BLUEBOXLib_API BLUEBOX_ErrorCodes __stdcall BLUEBOX_GenericCommand(BLUEBOX_Handle* Handle, char* Command, char* Reply);</pre>	

3 BlueBox Gen1 Functions Table

	BLUEBOX OEM LF	BLUEBOX OEM HF	BLUEBOX OEM HF E	BLUEBOX OEM UHF	BLUEBOX DESKTOP LF	BLUEBOX DESKTOP HF	BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL LF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL ACTIVE	BLUEBOX PORTAL UHF
BLUEBOX_GetSwRelease	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Init	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_End	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Open	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Close	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetAddress	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetChannel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetFwRelease	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Reset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetDateTime																			
BLUEBOX_SetDateTime																			
BLUEBOX_ReadParameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_WriteParameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_DefaultParameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_ReadSerialNumber																			
BLUEBOX_ReadMACAddress																	✓		
BLUEBOX_ReadConfiguration				✓								✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_WriteConfiguration				✓								✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_DefaultConfiguration				✓								✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_DataRequest	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_QueueRequest	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_FreeTagsMemory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_AllocateNotifyChannel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_DeallocateNotifyChannel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetNotification	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_FreeNotifyMemory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetOutput	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetReaderStatus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetTemperature																			
BLUEBOX_RfOnOff	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SelectiveRfOnOff																✓			

	BLUEBOX OEM LF	BLUEBOX OEM HF	BLUEBOX OEM HF E	BLUEBOX OEM UHF	BLUEBOX DESKTOP LF	BLUEBOX DESKTOP HF	BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL LF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL ACTIVE	BLUEBOX PORTAL UHF
BLUEBOX_ReadID_EM4305	✓				✓		✓	✓											
BLUEBOX_Write_EM4305	✓				✓		✓	✓											
BLUEBOX_ReadID_T5557	✓				✓		✓	✓											
BLUEBOX_Write_T5557	✓				✓		✓	✓											
BLUEBOX_ReadID_Q5	✓				✓		✓	✓											
BLUEBOX_WriteQ5	✓				✓		✓	✓											
BLUEBOX_ReadID_HITAG1	✓				✓		✓	✓											
BLUEBOX_ReadPage_HITAG1	✓				✓		✓	✓											
BLUEBOX_WritePage_HITAG1	✓				✓		✓	✓											
BLUEBOX_ReadID_HITAGS	✓				✓		✓	✓											
BLUEBOX_Write_HITAGS	✓				✓		✓	✓											
BLUEBOX_ReadPage_HITAGS	✓				✓		✓	✓											
BLUEBOX_WritePage_HITAGS	✓				✓		✓	✓											
BLUEBOX_ReadID_TITAN	✓				✓		✓	✓											
BLUEBOX_Reset_TITAN	✓				✓		✓	✓											
BLUEBOX_Login_TITAN	✓				✓		✓	✓											
BLUEBOX_WritePassword_TITAN	✓				✓		✓	✓											
BLUEBOX_SelectiveRead_TITAN	✓				✓		✓	✓											
BLUEBOX_SelectiveWrite_TITAN	✓				✓		✓	✓											
BLUEBOX_Inventory_ISO15693		✓				✓				✓	✓	✓	✓	✓					
BLUEBOX_ReadPage_ISO15693		✓				✓				✓	✓	✓	✓	✓					
BLUEBOX_ReadMultiPage_ISO15693												✓	✓						
BLUEBOX_WritePage_ISO15693		✓				✓				✓	✓	✓	✓	✓					
BLUEBOX_WriteMultiPage_ISO15693												✓	✓						
BLUEBOX_LockPage_ISO15693		✓				✓				✓	✓	✓	✓	✓					
BLUEBOX_Write_AFI_ISO15693												✓	✓	✓					
BLUEBOX_Lock_AFI_ISO15693												✓	✓	✓					
BLUEBOX_GetRandomNumber_ICODE_SLI_S		✓				✓													
BLUEBOX_SetPassword_ICODE_SLI_S		✓				✓													
BLUEBOX_WritePassword_ICODE_SLI_S		✓				✓													
BLUEBOX_LockPassword_ICODE_SLI_S		✓				✓													
BLUEBOX_64BitPasswordProtection_ICODE_SLI_S		✓				✓													
BLUEBOX_ProtectPage_ICODE_SLI_S		✓				✓													

	BLUEBOX OEM LF	BLUEBOX OEM HF	BLUEBOX OEM HF E	BLUEBOX OEM UHF	BLUEBOX DESKTOP LF	BLUEBOX DESKTOP HF	BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL LF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL ACTIVE	BLUEBOX PORTAL UHF
BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S		✓				✓													
BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S		✓				✓													
BLUEBOX_Destroy_SLI_S_ICODE_SLI_S		✓				✓													
BLUEBOX_EnablePrivacy_ICODE_SLI_S		✓				✓													
BLUEBOX_Inevntory_ISO14443A		✓	✓			✓				✓	✓								
BLUEBOX_ReadBlock_MIFARE_1k		✓	✓			✓				✓	✓								
BLUEBOX_WriteBlock_MIFARE_1k		✓	✓			✓				✓	✓								
BLUEBOX_ReadBlock_MIFARE_4k		✓	✓			✓				✓	✓								
BLUEBOX_WriteBlock_MIFARE_4k		✓	✓			✓				✓	✓								
BLUEBOX_ReadBlock_MIFARE_Ultralight		✓	✓			✓				✓	✓								
BLUEBOX_WriteBlock_MIFARE_Ultralight		✓	✓			✓				✓	✓								
BLUEBOX_ReadBlock_NTAG213		✓				✓													
BLUEBOX_WriteBlock_NTAG213		✓				✓													
BLUEBOX_ReadBlock_NTAG215		✓				✓													
BLUEBOX_WriteBlock_NTAG215		✓				✓													
BLUEBOX_ReadBlock_NTAG216		✓				✓													
BLUEBOX_WriteBlock_NTAG216		✓				✓													
BLUEBOX_Inventry_ISO14443B		✓				✓				✓	✓								
BLUEBOX_ReadBlock_SR176		✓				✓				✓	✓								
BLUEBOX_WriteBlock_SR176		✓				✓				✓	✓								
BLUEBOX_ReadRfParameters												✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_WriteRfParameters												✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Inventry_ISO18K6B																✓			
BLUEBOX_Read_ISO18K6B																✓			
BLUEBOX_Write_ISO18K6B																✓			
BLUEBOX_Inventry_ISO18K6C				✓											✓	✓	✓		
BLUEBOX_ProgramEPC_ISO18K6C																			
BLUEBOX_Read_ISO18K6C				✓											✓	✓	✓		
BLUEBOX_Write_ISO18K6C				✓											✓	✓	✓		
BLUEBOX_BlockWrite_ISO18K6C																			
BLUEBOX_Lock_ISO18K6C				✓											✓	✓	✓		
BLUEBOX_Kill_ISO18K6C				✓											✓	✓	✓		
BLUEBOX_FwUpgrade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	BLUEBOX OEM LF	BLUEBOX OEM HF	BLUEBOX OEM HF E	BLUEBOX OEM UHF	BLUEBOX DESKTOP LF	BLUEBOX DESKTOP HF	BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL LF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL HF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE QUAD CHANNEL	BLUEBOX INDUSTRIAL UHF LONG RANGE DUAL CHANNEL	BLUEBOX INDUSTRIAL ACTIVE	BLUEBOX PORTAL UHF
BLUEBOX_ReadNumberOfRegistrations																			✓
BLUEBOX_ReadOlderRegistration																			✓
BLUEBOX_CancelOlderRegistration																			✓
BLUEBOX_CancelAllRegistrations																			✓
BLUEBOX_ReadPreviousRegistration																			✓
BLUEBOX_GenericCommand	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4 BlueBox Gen2 Functions Table

	BLUEBOX GEN2 DESKTOP LF	BLUEBOX GEN2 DESKTOP HF	BLUEBOX GEN2 DESKTOP UHF	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL
BLUEBOX_GetSwRelease	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Init	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_End	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Open	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Close	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetAddress	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetChannel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetFwRelease	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_Reset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetDateTime				✓	✓	✓	✓	✓	✓				
BLUEBOX_SetDateTime				✓	✓	✓	✓	✓	✓				
BLUEBOX_ReadParameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_WriteParameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_DefaultParameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_ReadSerialNumber				✓	✓	✓	✓	✓	✓				
BLUEBOX_ReadMACAddress				✓	✓	✓	✓	✓	✓				
BLUEBOX_ReadConfiguration			✓	✓	✓	✓	✓	✓	✓				
BLUEBOX_WriteConfiguration			✓	✓	✓	✓	✓	✓	✓				
BLUEBOX_DefaultConfiguration			✓	✓	✓	✓	✓	✓	✓				
BLUEBOX_DataRequest	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_QueueRequest	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_FreeTagsMemory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_AllocateNotifyChannel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_DeallocateNotifyChannel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetNotification	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_FreeNotifyMemory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SetOutput				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetReaderStatus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_GetTemperature				✓	✓	✓	✓	✓	✓				

	BLUEBOX GEN2 DESKTOP LF	BLUEBOX GEN2 DESKTOP HF	BLUEBOX GEN2 DESKTOP UHF	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL
BLUEBOX_RfOnOff	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_SelectiveRfOnOff													
BLUEBOX_ReadID_EM4305	✓			✓	✓					✓			
BLUEBOX_Write_EM4305	✓			✓	✓					✓			
BLUEBOX_ReadID_T5557	✓			✓	✓					✓			
BLUEBOX_Write_T5557	✓			✓	✓					✓			
BLUEBOX_ReadID_Q5	✓			✓	✓					✓			
BLUEBOX_WriteQ5	✓			✓	✓					✓			
BLUEBOX_ReadID_HITAG1	✓			✓	✓					✓			
BLUEBOX_ReadPage_HITAG1	✓			✓	✓					✓			
BLUEBOX_WritePage_HITAG1	✓			✓	✓					✓			
BLUEBOX_ReadID_HITAGS	✓			✓	✓					✓			
BLUEBOX_Write_HITAGS	✓			✓	✓					✓			
BLUEBOX_ReadPage_HITAGS	✓			✓	✓					✓			
BLUEBOX_WritePage_HITAGS	✓			✓	✓					✓			
BLUEBOX_ReadID_TITAN	✓			✓	✓					✓			
BLUEBOX_Reset_TITAN	✓			✓	✓					✓			
BLUEBOX_Login_TITAN	✓			✓	✓					✓			
BLUEBOX_WritePassword_TITAN	✓			✓	✓					✓			
BLUEBOX_SelectiveRead_TITAN	✓			✓	✓					✓			
BLUEBOX_SelectiveWrite_TITAN	✓			✓	✓					✓			
BLUEBOX_Inventory_ISO15693		✓				✓	✓	✓			✓		
BLUEBOX_ReadPage_ISO15693		✓				✓	✓	✓			✓		
BLUEBOX_ReadMultiPage_ISO15693													
BLUEBOX_WritePage_ISO15693		✓				✓	✓	✓			✓		
BLUEBOX_WriteMultiPage_ISO15693													
BLUEBOX_LockPage_ISO15693		✓				✓	✓	✓			✓		
BLUEBOX_Write_AFI_ISO15693								✓					
BLUEBOX_Lock_AFI_ISO15693								✓					
BLUEBOX_GetRandomNumber_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_SetPassword_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_WritePassword_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_LockPassword_ICODE_SLI_S		✓				✓	✓				✓		

	BLUEBOX GEN2 DESKTOP LF	BLUEBOX GEN2 DESKTOP HF	BLUEBOX GEN2 DESKTOP UHF	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL
BLUEBOX_64BitPasswordProtection_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_ProtectPage_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_Destroy_SLI_S_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_EnablePrivacy_ICODE_SLI_S		✓				✓	✓				✓		
BLUEBOX_Inevntory_ISO14443A		✓				✓	✓				✓		
BLUEBOX_ReadBlock_MIFARE_1k		✓				✓	✓				✓		
BLUEBOX_WriteBlock_MIFARE_1k		✓				✓	✓				✓		
BLUEBOX_ReadBlock_MIFARE_4k		✓				✓	✓				✓		
BLUEBOX_WriteBlock_MIFARE_4k		✓				✓	✓				✓		
BLUEBOX_ReadBlock_MIFARE_Ultralight		✓				✓	✓				✓		
BLUEBOX_WriteBlock_MIFARE_Ultralight		✓				✓	✓				✓		
BLUEBOX_ReadBlock_NTAG213		✓				✓	✓				✓		
BLUEBOX_WriteBlock_NTAG213		✓				✓	✓				✓		
BLUEBOX_ReadBlock_NTAG215		✓				✓	✓				✓		
BLUEBOX_WriteBlock_NTAG215		✓				✓	✓				✓		
BLUEBOX_ReadBlock_NTAG216		✓				✓	✓				✓		
BLUEBOX_WriteBlock_NTAG216		✓				✓	✓				✓		
BLUEBOX_Inventory_ISO14443B		✓				✓	✓				✓		
BLUEBOX_ReadBlock_SR176		✓				✓	✓				✓		
BLUEBOX_WriteBlock_SR176		✓				✓	✓				✓		
BLUEBOX_ReadRfParameters			✓					✓	✓			✓	✓
BLUEBOX_WriteRfParameters			✓					✓	✓			✓	✓
BLUEBOX_Inventory_ISO18K6B													
BLUEBOX_Read_ISO18K6B													
BLUEBOX_Write_ISO18K6B													
BLUEBOX_Inventory_ISO18K6C			✓						✓			✓	✓
BLUEBOX_ProgramEPC_ISO18K6C													
BLUEBOX_Read_ISO18K6C			✓						✓			✓	✓
BLUEBOX_Write_ISO18K6C			✓						✓			✓	✓
BLUEBOX_BlockWrite_ISO18K6C													
BLUEBOX_Lock_ISO18K6C			✓						✓			✓	✓

	BLUEBOX GEN2 DESKTOP LF	BLUEBOX GEN2 DESKTOP HF	BLUEBOX GEN2 DESKTOP UHF	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL	BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL	BLUEBOX GEN2 BASIC UHF MID RANGE SINGLE CHANNEL
BLUEBOX_Kill_ISO18K6C			✓						✓			✓	✓
BLUEBOX_FwUpgrade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BLUEBOX_ReadNumberOfRegistrations													
BLUEBOX_ReadOlderRegistration													
BLUEBOX_CancelOlderRegistration													
BLUEBOX_CancelAllRegistrations													
BLUEBOX_ReadPreviousRegistration													
BLUEBOX_GenericCommand	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

5 BlueBox CX Functions Table

	BLUEBOX CX UHF MID RANGE SINGLE CHANNEL	BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL	BLUEBOX CX UHF LONG RANGE DUAL CHANNEL
BLUEBOX_GetSwRelease	✓	✓	✓
BLUEBOX_Init	✓	✓	✓
BLUEBOX_End	✓	✓	✓
BLUEBOX_Open	✓	✓	✓
BLUEBOX_Close	✓	✓	✓
BLUEBOX_SetAddress	✓	✓	✓
BLUEBOX_SetDevice	✓	✓	✓
BLUEBOX_GetDevice	✓	✓	✓
BLUEBOX_SetChannel	✓	✓	✓
BLUEBOX_GetFwRelease	✓	✓	✓
BLUEBOX_Reset	✓	✓	✓
BLUEBOX_GetDateTime	✓	✓	✓
BLUEBOX_SetDateTime	✓	✓	✓
BLUEBOX_ReadParameters	✓	✓	✓
BLUEBOX_WriteParameters	✓	✓	✓
BLUEBOX_DefaultParameters	✓	✓	✓
BLUEBOX_ReadSerialNumber	✓	✓	✓
BLUEBOX_ReadMACAddress	✓	✓	✓
BLUEBOX_ReadConfiguration	✓	✓	✓
BLUEBOX_WriteConfiguration	✓	✓	✓
BLUEBOX_DefaultConfiguration	✓	✓	✓
BLUEBOX_DataRequest	✓	✓	✓
BLUEBOX_QueueRequest	✓	✓	✓
BLUEBOX_FreeTagsMemory	✓	✓	✓
BLUEBOX_AllocateNotifyChannel	✓	✓	✓
BLUEBOX_DeallocateNotifyChannel	✓	✓	✓
BLUEBOX_GetNotification	✓	✓	✓
BLUEBOX_FreeNotifyMemory	✓	✓	✓
BLUEBOX_SetOutput	✓	✓	✓
BLUEBOX_GetReaderStatus	✓	✓	✓
BLUEBOX_GetTemperature	✓	✓	✓
BLUEBOX_RfOnOff	✓	✓	✓
BLUEBOX_SelectiveRfOnOff	✓	✓	✓

	BLUEBOX CX UHF MID RANGE SINGLE CHANNEL	BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL	BLUEBOX CX UHF LONG RANGE DUAL CHANNEL
BLUEBOX_ReadID_EM4305			
BLUEBOX_Write_EM4305			
BLUEBOX_ReadID_T5557			
BLUEBOX_Write_T5557			
BLUEBOX_ReadID_Q5			
BLUEBOX_WriteQ5			
BLUEBOX_ReadID_HITAG1			
BLUEBOX_ReadPage_HITAG1			
BLUEBOX_WritePage_HITAG1			
BLUEBOX_ReadID_HITAGS			
BLUEBOX_Write_HITAGS			
BLUEBOX_ReadPage_HITAGS			
BLUEBOX_WritePage_HITAGS			
BLUEBOX_ReadID_TITAN			
BLUEBOX_Reset_TITAN			
BLUEBOX_Login_TITAN			
BLUEBOX_WritePassword_TITAN			
BLUEBOX_SelectiveRead_TITAN			
BLUEBOX_SelectiveWrite_TITAN			
BLUEBOX_Inventory_ISO15693			
BLUEBOX_ReadPage_ISO15693			
BLUEBOX_ReadMultiPage_ISO15693			
BLUEBOX_WritePage_ISO15693			
BLUEBOX_WriteMultiPage_ISO15693			
BLUEBOX_LockPage_ISO15693			
BLUEBOX_Write_AFI_ISO15693			
BLUEBOX_Lock_AFI_ISO15693			
BLUEBOX_GetRandomNumber_ICODE_SLI_S			
BLUEBOX_SetPassword_ICODE_SLI_S			
BLUEBOX_WritePassword_ICODE_SLI_S			
BLUEBOX_LockPassword_ICODE_SLI_S			
BLUEBOX_64BitPasswordProtection_ICODE_SLI_S			
BLUEBOX_ProtectPage_ICODE_SLI_S			
BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S			
BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S			

	BLUEBOX CX UHF MID RANGE SINGLE CHANNEL	BLUEBOX CX UHF LONG RANGE SINGLE CHANNEL	BLUEBOX CX UHF LONG RANGE DUAL CHANNEL
BLUEBOX_Destroy_SLI_S_ICODE_SLI_S			
BLUEBOX_EnablePrivacy_ICODE_SLI_S			
BLUEBOX_Inevntory_ISO14443A			
BLUEBOX_ReadBlock_MIFARE_1k			
BLUEBOX_WriteBlock_MIFARE_1k			
BLUEBOX_ReadBlock_MIFARE_4k			
BLUEBOX_WriteBlock_MIFARE_4k			
BLUEBOX_ReadBlock_MIFARE_Ultralight			
BLUEBOX_WriteBlock_MIFARE_Ultralight			
BLUEBOX_Inventory_ISO14443B			
BLUEBOX_ReadBlock_SR176			
BLUEBOX_WriteBlock_SR176			
BLUEBOX_ReadRfParameters	✓	✓	✓
BLUEBOX_WriteRfParameters	✓	✓	✓
BLUEBOX_Inventory_ISO18K6B			
BLUEBOX_Read_ISO18K6B			
BLUEBOX_Write_ISO18K6B			
BLUEBOX_Inventory_ISO18K6C	✓	✓	✓
BLUEBOX_ProgramEPC_ISO18K6C	✓	✓	✓
BLUEBOX_Read_ISO18K6C	✓	✓	✓
BLUEBOX_Write_ISO18K6C	✓	✓	✓
BLUEBOX_BlockWrite_ISO18K6C	✓	✓	✓
BLUEBOX_Lock_ISO18K6C	✓	✓	✓
BLUEBOX_Kill_ISO18K6C	✓	✓	✓
BLUEBOX_FwUpgrade	✓	✓	✓
BLUEBOX_ReadNumberOfRegistrations			
BLUEBOX_ReadOlderRegistration			
BLUEBOX_CancelOlderRegistration			
BLUEBOX_CancelAllRegistrations			
BLUEBOX_ReadPreviousRegistration			
BLUEBOX_GenericCommand	✓	✓	✓

6 Document Revision History

Revision	Date	Description
1.00	30/04/10	First release.
1.01	05/05/10	<p>Changes in supported readers list to add the new readers managed from the library release 2.0.0.</p> <p>Added the document revision history section.</p> <p>Added definitions, functions, enums and structs to manage the new readers.</p> <p>Changes in BLUEBOX_SetChannel function parameters.</p> <p>Changes in nibble coding and gain enums definitions (ref. LF readers).</p> <p>Changes in BLUEBOX_GeneralParameters struct definition.</p> <p>Changes in BLUEBOX_TagType enum definition.</p> <p>Changes in MIFARE key enum definitions (ref. HF readers).</p> <p>Changes in BLUEBOX_ReaderStatus struct definition (deleted the Line flag in BLUEBOX INDUSTRIAL readers definition).</p>
1.02	18/06/10	<p>Changes in supported readers list to add the new readers managed from the library release 3.0.0.</p> <p>Added the 'spontaneous' message notifications by adding the functions:</p> <ul style="list-style-type: none"> • BLUEBOX_AllocateNotifyChannel • BLUEBOX_DeallocateNotifyChannel • BLUEBOX_GetNotification <p>and the error codes:</p> <ul style="list-style-type: none"> • BLUEBOX_AllocationError <p>and the structs:</p> <ul style="list-style-type: none"> • BLUEBOX_Notify <p>Added functions to manage the device type:</p> <ul style="list-style-type: none"> • BLUEBOX_SetDevice • BLUEBOX_GetDevice <p>Added the ISO 15693 AFI (Application Family Identifier) management in BLUEBOX_Inventory_ISO15693 function.</p> <p>Added functions to manage ISO 15693 AFI (Application Family Identifier):</p> <ul style="list-style-type: none"> • BLUEBOX_Write_AFI_ISO15693 • BLUEBOX_Lock_AFI_ISO15693 <p>Improved BLUEBOX_Close and BLUEBOX_End functions. The BLUEBOX_End function implicitly calls the BLUEBOX_Close function.</p> <p>Changes in BLUEBOX_SetChannel function parameters strings, added the retransmission numbers at the end of Settings string with RS232/RS485 interface.</p> <p>Added the BLUEBOX_Tag struct definition.</p> <p>Changes in BLUEBOX_RfParameters struct definition.</p> <p>Added the upgrade firmware by adding the functions:</p> <ul style="list-style-type: none"> • BLUEBOX_FwUpgrade • BLUEBOX_GetUpgradeStatus

		<p>and the error codes:</p> <ul style="list-style-type: none"> • BLUEBOX_FileError <p>and the enums:</p> <ul style="list-style-type: none"> • BLUEBOX_UpgReader <p>Changes in definitions to uniform the code writing rules.</p> <p>Added the library release information in document revision history table.</p>
1.03	08/07/10	<p>Corrections in the document revision history in description of the 1.02 document release.</p> <p>Changes in supported readers list to add the new readers managed from the library release 4.0.0.</p> <p>Changes in BLUEBOX_SetChannel function parameters strings, added the communication timeout of Settings string with all interfaces.</p> <p>Changes in BLUEBOX_GetFwRelease to allow the auxiliary reader's fw version reading.</p> <p>Changes in configuration functions:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadParameters; • BLUEBOX_WriteParameters; • BLUEBOX_ReadRfParameters; • BLUEBOX_WriteRfParameters; <p>parameters to make them more flexible. Also deleted all the enumerations and structures related to them.</p> <p>Changes in BLUEBOX_GetReaderStatus function parameters to make it more flexible. Also deleted all the enumerations and structures related to it.</p> <p>Changed the BLUEBOX_UpgReader numeration to BLUEBOX_Reader and also changed its items names.</p> <p>Added the BLUEBOX_TagError error code in the following functions:</p> <ul style="list-style-type: none"> • BLUEBOX_Inventory_ISO15693; • BLUEBOX_Inventory_ISO14443A; • BLUEBOX_Inventory_ISO14443B.
1.04	02/08/10	<p>Changed the BLUEBOX_GetDevice and BLUEBOX_SetDevice function parameters by adding the firmware version major and minor numbers to manage different features in different firmware versions.</p>
1.05	05/08/10	<p>Added the EM4305 and T5557 tags management in BLUEBOX INDUSTRIAL LF SHORT RANGE SINGLE/DUAL CHANNEL, BLUEBOX OEM LF and BLUEBOX DESKTOP LF by adding, in BLUEBOX_TagType the enumerators:</p> <ul style="list-style-type: none"> • BLUEBOX_EM4305; • BLUEBOX_T5557; <p>and definitions:</p> <ul style="list-style-type: none"> • BLUEBOX_EM4305_ID_SIZE; • BLUEBOX_T5557_ID_SIZE; <p>and functions:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadID_EM4305; • BLUEBOX_Write_EM4305; • BLUEBOX_ReadID_T5557;

		<ul style="list-style-type: none"> • BLUEBOX_Write_T5557.
1.06	05/10/10	<p>Corrections in the document revision history in description of the 1.04 document release.</p> <p>Deleted the Antenna parameter from BLUEBOX_RfOnOff function.</p> <p>Deleted the BLUEBOX_GetUpgradeStatus function.</p> <p>Added the BLUEBOX PORTAL UHF reader management. The affected functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_SetDevice; • BLUEBOX_GetDevice; • BLUEBOX_GetFwRelease; • BLUEBOX_ReadParameters; • BLUEBOX_WriteParameters; • BLUEBOX_DefaultParameters; • BLUEBOX_GetReaderStatus; • BLUEBOX_ReadRfParameters; • BLUEBOX_WriteRfParameters; <p>the functions added are:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadNumberOfRegistrations; • BLUEBOX_ReadOlderRegistration; • BLUEBOX_CancelOlderRegistration; • BLUEBOX_CancelAllRegistrations; • BLUEBOX_ReadPreviousRegistration; <p>the affected definitions / enumerations / structures are:</p> <ul style="list-style-type: none"> • BLUEBOX_ErrorCodes; <p>the definitions / enumerations / structures added are:</p> <ul style="list-style-type: none"> • BLUEBOX_Input; • BLUEBOX_Registration.
1.07	17/01/11	<p>Added the ISO 18000-6C (EPC C1G2) with variable UID size tags management. The affected definitions are:</p> <ul style="list-style-type: none"> • BLUEBOX_ISO18K6C_UID_SIZE. <p>Added the section remarks in functions:</p> <ul style="list-style-type: none"> • BLUEBOX_GetFwRelease; • BLUEBOX_ReadParameters.
1.08	09/09/11	<p>Added the firmware release related to this technical manual in the first page.</p> <p>Added the x64 architecture support in section 1.</p> <p>Deleted the BLUEBOX INDUSTRIAL UHF SHORT RANGE SINGLE CHANNEL reader management (replaced with the MID RANGE one).</p> <p>Added the customization management through the variant management. The affected functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_SetDevice; • BLUEBOX_GetDevice. <p>Changed the BLUEBOX_AllocateNotifyChannel parameters and function prototype.</p>

		Changes in section 'Document Revision History' (this section).
1.09	24/10/11	<p>Extended the BLUEBOX_Input enumeration to support the 'no input' case.</p> <p>Increased the maximum tag's ID length supported (BLUEBOX_MAX_ID_LENGTH definition).</p> <p>Changes to BLUEBOX_Tag and BLUEBOX_Notify structures.</p>
1.10	19/01/12	<p>Deleted the maximum tag's ID length definitions (BLUEBOX_MAX_ID_LENGTH).</p> <p>Changed the tag's ID management from static array to dynamic array. The affected structures are:</p> <ul style="list-style-type: none"> • BLUEBOX_Tags; • BLUEBOX_Notify; • BLUEBOX_Registration. <p>The affected functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_FreeTagsMemory; • BLUEBOX_FreeNotifyMemory.
1.11	22/02/12	<p>Added the ICODE SLI-S tag management in BLUEBOX DESKTOP HF and BLUEBOX OEM HF by adding, in BLUEBOX_TagType the enumerators:</p> <ul style="list-style-type: none"> • BLUEBOX_ICODE_SLI_S; <p>and enumerations:</p> <ul style="list-style-type: none"> • BLUEBOX_ICODE_SLI_S_PasswordIdentifier; • BLUEBOX_ICODE_SLI_S_ProtectionStatus; <p>and definitions:</p> <ul style="list-style-type: none"> • BLUEBOX_ICODE_SLI_S_RND_SIZE; • BLUEBOX_ICODE_SLI_S_PWD_SIZE; <p>and structures:</p> <ul style="list-style-type: none"> • BLUEBOX_ICODE_SLI_S_BlockProtectionStatus; <p>and functions:</p> <ul style="list-style-type: none"> • BLUEBOX_GetRandomNumber_ICODE_SLI_S; • BLUEBOX_SetPassword_ICODE_SLI_S; • BLUEBOX_WritePassword_ICODE_SLI_S; • BLUEBOX_LockPassword_ICODE_SLI_S; • BLUEBOX_64BitPasswordProtection_ICODE_SLI_S • BLUEBOX_ProtectPage_ICODE_SLI_S; • BLUEBOX_LockPageProtectionCondition_ICODE_SLI_S; • BLUEBOX_GetMultipleBlockProtectionStatus_ICODE_SLI_S; • BLUEBOX_Destroy_SLI_S_ICODE_SLI_S; • BLUEBOX_EnablePrivacy_ICODE_SLI_S;
1.12	10/10/12	<p>Added the BLUEBOX Gen2 readers support to the library (BLUEBOX GEN2 DESKTOP LF, BLUEBOX GEN2 DESKTOP HF, BLUEBOX GEN2 DESKTOP UHF, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL LF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF SHORT RANGE DUAL CHANNEL, BLUEBOX GEN2 INDUSTRIAL HF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 INDUSTRIAL UHF MID RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC LF SHORT RANGE SINGLE CHANNEL, BLUEBOX</p>

		<p>GEN2 BASIC HF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC UHF SHORT RANGE SINGLE CHANNEL, BLUEBOX GEN2 BASIC MID RANGE SINGLE CHANNEL.</p> <p>Deleted the list of the supported readers in section 1 and added sections 4 and 5 with the readers supported functions tables.</p> <p>Added the management of a second auxiliary reader. The affected enumerations are:</p> <ul style="list-style-type: none"> • BLUEBOX_Reader. <p>The affected functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_GetFwRelease; • BLUEBOX_FwUpgrade. <p>Added the BLUEBOX_Reset function.</p> <p>Added the management of the configuration pages of the readers. The added functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadConfiguration; • BLUEBOX_WriteConfiguration; • BLUEBOX_DefaultConfiguration. <p>Removed remarks in BLUEBOX_SetDevice and BLUEBOX_GetDevice functions.</p> <p>Added the BLUEBOX_GenericCommand function.</p>
1.13	29/04/13	<p>Added the BLUEBOX_GetTemperature function.</p> <p>Added the management of the HITAG 1 transponders. The affected enumerations are:</p> <ul style="list-style-type: none"> • BLUEBOX_TagType (added BLUEBOX_HITAG_1). <p>The added definitions are:</p> <ul style="list-style-type: none"> • BLUEBOX_HITAG1_ID_SIZE; • BLUEBOX_HITAG1_PAGE_SIZE. <p>The added functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadID_HITAG1; • BLUEBOX_ReadPage_HITAG1; • BLUEBOX_WritePage_HITAG1.
1.14	20/05/13	<p>Changed the BLUEBOX_ISO18K6C_UID_SIZE to 66 bytes.</p> <p>Added the read/write multi page of ISO 15693 tags. The added functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadMultiPage_ISO15693; • BLUEBOX_WriteMultiPage_ISO15693. <p>Added the management of BLUEBOX GEN2 OEM UHF readers.</p>
1.15	08/10/13	<p>Added the BLUEBOX OEM UHF reader support to the library.</p> <p>Added the enumeration items</p> <ul style="list-style-type: none"> • BLUEBOX_EM4305; • BLUEBOX_T5557; • BLUEBOX_ICODE_SLI_S; • BLUEBOX_HITAG_1; • BLUEBOX_MIFARE_MINI; • BLUEBOX_MIFARE_DESFIRE;

		<ul style="list-style-type: none"> • BLUEBOX_MIFARE_7BUID_2k; • BLUEBOX_MIFARE_7BUID_4k; • BLUEBOX_MIFARE_PLUS_2k; • BLUEBOX_MIFARE_PLUS_4k; • BLUEBOX_SRI512; • BLUEBOX_JCOS; • BLUEBOX_PICOPASS; <p>to BLUEBOX_TagType enumeration.</p> <p>Added the definitions:</p> <ul style="list-style-type: none"> • BLUEBOX_MIFARE_DESFIRE_UID_SIZE; • BLUEBOX_MIFARE_7BUID_2k_UID_SIZE; • BLUEBOX_MIFARE_7BUID_4k_UID_SIZE; • BLUEBOX_MIFARE_PLUS_2k_UID_SIZE; • BLUEBOX_MIFARE_PLUS_4k_UID_SIZE; • BLUEBOX_SRI512_UID_SIZE; • BLUEBOX_JCOS_UID_SIZE; • BLUEBOX_PICOPASS_UID_SIZE. <p>Added the functions:</p> <ul style="list-style-type: none"> • BLUEBOX_GetDateTime; • BLUEBOX_SetDateTime; • BLUEBOX_ReadSerialNumber; • BLUEBOX_ReadMACAddress; • BLUEBOX_SelectiveRfOnOff. <p>Updated the supported commands table.</p>
1.16	10/02/15	<p>Added the BLUEBOX CX UHF reader support to the library.</p> <p>Added the functions:</p> <ul style="list-style-type: none"> • BLUEBOX_ProgramEPC_ISO18K6C; • BLUEBOX_BlockWrite_ISO18K6C. <p>Updated the supported commands tables.</p>
1.17	29/09/16	<p>Added the management of the NTAG213/215216/ transponders. The added definitions are:</p> <ul style="list-style-type: none"> • BLUEBOX_NTAG21x_UID_SIZE; • BLUEBOX_NTAG21x_BLOCK_SIZE. <p>The added functions are:</p> <ul style="list-style-type: none"> • BLUEBOX_ReadBlock_NTAG213; • BLUEBOX_WriteBlock_NTAG213; • BLUEBOX_ReadBlock_NTAG215; • BLUEBOX_WriteBlock_NTAG215; • BLUEBOX_ReadBlock_NTAG216; • BLUEBOX_WriteBlock_NTAG216.
1.18	30/09/25	Format changes in all sections and corrections of minor mistakes.