SM130
SM132-USB

Software Development Kit - SDK
USER MANUAL

EVALUATION / DEVELOPMENT KIT

For a fast starting and a product just in days, we recommend you to buy one of our development/evaluation kits. SDK – Software Development Kit, is included freely in Deluxe versions of the kits or it can be purchased separately.

For Online Shopping, you can visit:
http://www.sonmicro.com/shop/shop3.php

Figure 1 – SM1013 Evaluation Kit – Deluxe

Figure 2 – SM132-USB – USB Mifare Reader

1. INTRODUCTION

This document explains usage of Mifare ActiveX component distributed with the Software Development Kit (SDK). Users can quickly add Mifare functions to existing software or create new software for Mifare applications easily with the provided library.

ActiveX library provides high level APIs to communicate with the supported devices (See Section 1.1 for the supported devices) and useful functions. Users never need to know about the communication protocol occurring between the device and the computer, ActiveX library will handle with that. Communication channel is based on Com Port of the computer and ActiveX component can also reliably be used with virtual com port or the USB-Serial converters.

It is strongly recommended for users who are strange to Mifare, first read about Mifare basics. Brief information for Mifare and its application can be found in User Manual file at our web page. http://www.sonmicro.com/1356/d1356.php

Any software development environment that supports Windows ActiveX components can be used to develop your Mifare application with the SonMicro Mifare ActiveX component. Followings are the examples for popular software development environments that can use Mifare ActiveX component.

- Delphi 4, Delphi 5, Delphi 6, Delphi 7, Delphi .Net
- Visual Basic, Visual C++,
- Visual Basic .NET, Visual C# .Net, Visual C++ .Net
- etc.

Operating systems other than Microsoft Windows is not supported currently.

1.1 Supported Devices

Mifare SDK supports the following devices:

- SM130 Mifare Module
- SM132-USB Mifare Reader (Integrated with Serial-to-USB interface and PCB antenna)

Supported Development kits:

- SM1013 Eval. kit for SM130
- SM1013USB Eval. Kit for SM132-USB

Please note that SM132-USB module is connected over USB interface but the control of these modules is still done in "classic serial port" manner with the created virtual com port. Driver for SM132USB can be downloaded at our web site.

1.2 How to Register ActiveX DLL

ActiveX (*.ocx) file should needs to be registered before using it.
Registering ActiveX DLL is very easy.

ActiveX Library, SMX10.ocx comes with SDK. To Register the SMX10.ocx file;

- Go to Start>Run
- Type regsvr32 path to the ocx file.

  If you copy/move the ocx file into “windows/system “ directory then you do not need to write whole path.

  Ex: regsvr32 SMX10.ocx

  It is not required to move ocx file to system directory. In this case write complete path.

  Ex: regsvr32 c:\myactivex\SMX10.ocx

To unregister the ActiveX Control, use “/u” switch

Ex: regsvr32 /u c:\myactivex\SMX10.ocx

1.3 Example VB 6.0 Project

There is an example Visual Basic 6 project comes with the SDK. Before using this project, SMX10.ocx should needs to be registered to the Windows OS. Project file is not illustrating a professional software but simple commands to illustrate communication between the Mifare module with ActiveX Control.

To quickly validate Mifare reader and ActiveX Control;

- Register ActiveX Control as explained above
- Open Project file in Visual Basic, or run Project exe file
- Select the right com port
- Open the com port
- Click on the Reset Button
- You should be able to See Firmware Version string in the status bar

To Read a block quickly;

- Open the com port
- Click Enable/Disable Auto Mode to enable the Auto Mode
- Select Mifare Block to be read
- Place a tag near the antenna
- Click on the Read Button

2. TYPE STRUCTURES

2.1 TS_Auth

This type structure can be used to pass parameters for Authentication command.

Elements of TS_Auth Structure

```
{
      Source              :Byte
      Key_Type            :Byte
      Keys                :TS_Keys
      E2prom_Block_No     :Byte
}
```

Quick Example for VB:

```
Dim My_auth as TS_Auth

My_auth.Source        = Provided_Key        ‘ Provided_Key = 1
My_auth.Key_Type      = TypeA               ‘ TypeA = 0
My_auth.Keys.Key[0]   = &H30                ‘ Key is 30 45 56 73 8B D0
My_auth.Keys.Key[1]   = &H45
My_auth.Keys.Key[2]   = &H56
My_auth.Keys.Key[3]   = &H73
My_auth.Keys.Key[4]   = &H8B
My_auth.Keys.Key[5]   = &HD0
```

Source:
This paramater determines the authentication source. There are three available auhentication sources. User can use defined constants/enumarators or manually enter the value to select the authentication source.

Valid Input/Value	Enumarator / Defined Constant
0	Mifare _Default (1*)
1	Provided_Key (2*)
2	E2Prom_Block_No (3*)

Notes: 1* If Source value is 0 then, Key TypeA and default key FF FF FF FF FF FF will be used to authenticate. Key_Type, Keys and E2prom_Block_No paramaters will not have any influence on the authentication command

2* If source value is 1 then elements: Key_Type and Keys in TS_Auth type structrure will be used to authenticate.

3* If Source value is 2 then elements: Key_Type and E2prom_Block_No will be used to authentiate. There are 16 programmable Keys each for TypeA and TypeB in the device internal memory. These keys are organized in 16 blocks and user can select the key with block number without revealing the Key in the communication channel.

Key Type:
If the Source value is 1 or 2, this parameter will be used to determine key type when authenticating.

Valid Input/Value	Enumarator / Defined Constant
0	TypeA (1*)
1	TypeB (2*)

Keys:
If the Source value is 1 or 2, this parameter will be used to determine the key to access when authenticating. Keys is defined as TS_Keys which is a 6 byte array.

E2prom Block No
If the Source value is 2, this parameter will be used to determine the location number of internal memory block in which the key, to be used to authenticate, is programmed.

Valid Input/Value	Enumarator / Defined Constant
0	E2prom_Block0
1	E2prom_Block1
...	...
15	E2prom_Block15

2.2 TS_Keys

This type structure can be used to pass the parameters of e2prom key.

Elements of TS_Keys Structure
{
Key :Byte[6]
}

Quick Example for VB:

```
Dim My_keys as TS_Keys

My_keys.Key[0] = &H30                ‘ Key is 30 45 56 73 8B D0
My_keys.Key[1] = &H45
My_keys.Key[2] = &H56
My_keys.Key[3] = &H73
My_keys.Key[4] = &H8B
My_keys.Key[5] = &HD0
```

Key:
Key will be used when programming internal e2prom as the Master Key. Key is defined as 6 byte array.

2.3 TS_Mifare_Block Structure

Mifare_ Block structure is used to determine to get or set the bytes of Mifare Block.

Elements of TS_Mifare_Block Structure
{
Bytes :Byte[16]
}

Quick Example for VB:

```
Dim My_block as TS_Mifare_Block
Dim i As Byte

For i=0 to 15
       My_block.Bytes(i)     = &HFF
Next i
```

Bytes:
Bytes is used to hold the values of the bytes of Mifare Block, defined as 16 bytes of array.

2.4 TS_Output_Pins Structure

This type is used to set the output pins of the Mifare module.

Elements of TS_Output_Pins Structure
{
Value :Byte
}

Quick Example for VB:

```
Dim My_Pins as TS_Output_Pins

My_Pins.Value = Output_2HIGH_Output1HIGH    'Value of the Output is 3
```

Value:
Value is used to determine the level of the output pins. It is defined as Byte.

Valid Input/Value	Enumarator / Defined Constant
0	Output_2LOW_Output1LOW
1	Output_2LOW_Output1HIGH
2	Output_2HIGH_Output1LOW
3	Output_2HIGH_Output1HIGH

2.5 TS_Access_Bytes Structure

This type structure can be used to pass the parameters for write sector trailer command or to get the parameters for create access bytes command.

Elements of TS_Access_Bytes Structure
{
Byte6 :Byte
Byte7 :Byte
Byte8 :Byte
User_Data :Byte
}

Quick Example for VB:

```
Dim My_Access as TS_Access_Bytes

My_Access.Byte6 = &HFF                    'Access_Bytes FF 07 80 69
My_Access.Byte7 = &H07
My_Access.Byte8 = &H80
My_Access.User_Data = &H69
```

Byte6, Byte7, Byte8:
This type structure is only used to write sector trailer block. Byte6, Byte7, Byte8 will be used to determine for accessing the Blocks of that Sector. They are defined as Byte.

User Data:
User Data byte is the 9. byte of the sector trailer block and it is used to save a byte which sends by user.

2.6 TS_UL_Otp Structure

This type structure can be used to pass the parameters for write OTP command.

Elements of TS_UL_Otp Structure
{
Bytes :Byte[4]
}

Quick Example for VB:

```
Dim My_Otp as TS_UL_Otp

For i=0 to 3
       My_Otp.Bytes(i) = &H00                    'Bytes 00 00 00 00
Next i
```

Bytes:

2.7 TS_UL_Lock Structure

This type structure can be used to pass the parameters for write OTP command.

Elements of TS_UL_Lock Structure

```
{
      Bytes                    :Byte[2]
}
```

Quick Example for VB:

```
Dim My_Lock as TS_UL_Otp

For i=0 to 1
       My_Lock.Bytes(i) = &H00                    'Bytes 00 00 00 00
Next i
```

3. PROCEDURES AND FUNCTIONS

It is assumed that developer has learn the basics of Mifare and the SonMicro Mifare device. Please visit http://www.sonmicro.com/1356/d1356.php to see useful documents and software to get a fast understanding for Mifare and mifare applications. For SonMicro Mifare Readers details please reference the relevant product's User Manual documents.

ActiveX control was written in Delphi, therefore Delphi/Pascal representations will be used to for the function prototypes. Examples to access ActiveX functions will be given for Visual Basic. Please visit http://www.sonmicro.com to check for new examples or code samples for different development environments.

Hints for Non-Delphi Developers:

A parameter inside the paranthesis mean argument will be passed to the ActiveX function.

Function CMD_myfunction(myvalue:Widestring):integer;

Myvalue as Widestring will be passed to the CMD_myfunction. Function will return integer value.

A parameter inside the paranthesis with "var" keyword means argument wil be passed to the ActiveX function whether it is used or not by the ActiveX function, and then ActiveX function may modify or update the content or overwrite to the parameter. Var is actually pointing to the variable.

Function CMD_Get_ComPorts(var comports:Widestring):integer;

3.1 CMD_Get_ComPorts

This function, get the list of available com ports, if exist, in string format.

function TSMifareX.CMD_Get_ComPorts (**Var** comports: **WideString**): **Integer;**	
Arguments	Comports: WideString * Com port names are returned as comma sepaperated in this variable.
Returns	0 : No comport available Others : Number of comports

Example for VB:

```
Dim com_count as integer
Dim comport_list as String

com_count = SMifareX1.CMD_Get_ComPorts(comport_list)
‘Com Port names can be found in comport_list variable as comma seperated
‘format
```

3.2 CMD_Get_Delimineted

This function, will get the parameter/Value that is delimineted with a deliminator string. For each call of this function next parameter will be retrieved and the retrieved parameter will be deleted from the original string.

function TSMifareX.CMD_Get_Delimineted (**Const** Deliminator: **WideString;** **Var** Value: **Widestring**; **Var** Org_Str: **Widestirng**;): **Integer;**	
Arguments	Deliminator: WideString Value:WideString; Org_Str: WideString; *Deliminator can be any string, that will be searched in the Org_Str *Value is the parameter retrieved from Org_Str that is delimineted with Deliminator *Org_Str is the string that search process will be done in.
Returns	0 : Parameter is found in the Org_Str 1 : Parameter is not found in the Org_Str

Example for VB:

```
Dim Status As integer
Dim Param As String
Dim Comport_List As String;

Status = SMifareX1.CMD_Get_Delimineted(",", Param, Comport_List)
'Note that Comport_List string will get shorter until all params are retrieved.
‘Ex: Comport_List is ”COM4,COM7,”
‘When CMD_Get_Delimineted is called first time;
‘Param will be “COM4” and the Comport_List will be “COM7,”
‘If you call the function again;
‘Param will be “COM7” and the Comport_List will be “”(empty)
```

3.3 CMD_OpenPort

This function creates and opens the selected ComPort.

function TSMifareX.CMD_OpenPort(const PortName, Baud**: WideString**): **Integer**	
Arguments	Portname: WideString Baud : WideString
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim Port As String
Dim Baud As String
Dim error_code As Long

Baud = “19200”
Port = “COM5”

error_code = SMifareX1.CMD_OpenPort(Port, Baud)
'Create and open COM5 with 19200 baud rate
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.4 CMD_ClosePort

This function close and destroy the ComPort, if it was opened previously.

function TSMifareX.CMD_ClosePort( ): **Integer**	
Arguments	No Arguments
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long

error_code = SMifareX1.CMD_ClosePort
'Close and destroy Com if opened before
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.5 CMD_AutoMode

This function enables/disables Auto Mode. When Auto mode is enabled, Halt, Select Tag, and Authenticate commands will run automatically before any Read/Write operation.

function TSMifareX.CMD_Reset(var Mode: Byte; var Auth_Params: TS_Auth): **Integer**	
Arguments	Mode: Byte Auth_Params: TS_Auth * Mode can be 0 or 1 . 0 Disables, and 1 enables Auto Mode * Auth_Params include necessary Authentication parameters and configuration.
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim my_auth As TS_Auth

    my_auth.Source = Provided_Key              'Provided Key Authentication
    my_auth.Key_Type = Type_A                  'Key Type
    my_auth.Keys.Key(0) = "&H" + Text1.Text                              'Key
    my_auth.Keys.Key(1) = "&H" + Text2.Text
    my_auth.Keys.Key(2) = "&H" + Text3.Text
    my_auth.Keys.Key(3) = "&H" + Text4.Text
    my_auth.Keys.Key(4) = "&H" + Text5.Text
    my_auth.Keys.Key(5) = "&H" + Text6.Text

    error_code = SMifareX1.CMD_AutoMode(1, my_auth)                'Auto mode On
```

3.6 CMD_Halt

This function halts the PICC.

function TSMifareX.CMD_Halt( ): **Integer**	
Arguments	No Arguments
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long

error_code = SMifareX1.CMD_Halt
'Halts the PICC
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.7 CMD_Reset

This function resets the Mifare module. Firmware version string passes into the Firmware string variable.

function TSMifareX.CMD_Reset(var Firmware_Version: WideString): **Integer**	
Arguments	Firmware Version : Widestring * Module sends the firmware version after reset in ascii format
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim firmware As String

error_code = SMifareX1.CMD_Reset(firmware)
'Resets the module, firmware version can be found in firmware arg.
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.8 CMD_SelectTag

This function selects a tag, if the tag is in RF field.

function TSMifareX.CMD_SelectTag(var Tag_TypeSerial: WideString): **Integer**	
Arguments	Tag_TypeSerial : Widestring * Tag_TypeSerial represents the tag type and the serial number of tag in hex-string format
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim My_Tag As String

error_code = SMifareX1.CMD_SelectTag(My_Tag)
‘Selects a tag, returns the tag type and serial number in hex-string format
‘in My_Tag argument.
‘My_Tag = 020467DC31 (02 is the tag type/ 31 DC 67 04 is serial number)
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.9 CMD_SeekTag

This function seeks a tag, and selects a tag as soon as the tag presents in RF field. When this command is used, the expected response should be retrieved in OnPacketReceive event(See Section 3.32)

function TSMifareX.CMD_SeekTag( ): **Integer**	
Arguments	No Arguments
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long

error_code = SMifareX1.CMD_SeekTag()
‘Seeks a Tag, and selects it when the tag presents in RF field
‘So the serial number is coming OnReceive event of the module
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.10 CMD_Authenticate

This function authenticates the tag, with the specified authentication parameters.

function TSMifareX.CMD_Authneticate(var Block_No: Byte; var Auth_Params: TS_Auth): **Integer**	
Arguments	Block_No : Byte Auth_Params : TS_Auth * Block_No represents the blok number of the tag * Auth_Params represents all authentication parameters in itself
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim my_Auth As TS_Auth
Dim Block_No As Byte

Block_No = 2
my_auth.Source = Provided_Key        'Provided Key is the Source of the Auth
my_auth.Key_Type = Type_A            'Key Type is A
my_auth.Keys.Key(0) = &HFF           'Key is FF FF FF FF FF FF
my_auth.Keys.Key(1) = &HFF
my_auth.Keys.Key(2) = &HFF
my_auth.Keys.Key(3) = &HFF
my_auth.Keys.Key(4) = &HFF
my_auth.Keys.Key(5) = &HFF

error_code = SMifareX1.CMD_Authenticaiton(Block_No, My_Auth)
'Second Block is authenticated with my_auth parameters
'if error_code is not zero look at the error code table in order to understand
'the error
```

3.11 CMD_FirmwareVersion

This function reads the firmware of the module.

function TSMifareX.CMD_FirmwareVersion(var Firmware_Version: WideString): **Integer**	
Arguments	Firmware Version : WideString * Firmware version of the module returns in this variable with hex-string forrmat
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim my_firmware As String

error_code = SMifareX1.CMD_FirmwareVersion(my_firmware)
‘Reads the firmware of the module
‘my_firmware = 554D20312E30 → UM 1.0
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.12 CMD_ReadInputPins

Reads the input pins situation.

function TSMifareX.CMD_ReadInputPins(var Input_Status: WideString): **Integer**	
Arguments	Input Status : WideString * Input Status returns the pins situation (0 is low, 1 is high)
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Input_Status As String

error_code = SMifareX1.CMD_ReadInputPins(Input_Status)
‘Reads the input pins
‘Input_Status = 00 → all input pins are low
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.13 CMD_WriteOutputPins

Write the value to the output pins

function TSMifareX.CMD_WriteOutputPins(var Output_Value: TS_Output_Pins): **Integer**	
Arguments	Output_Value: TS_Output_Pins * Output value sets the output pins
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Output_Status As TS_Output_Pins

Output_Status.Value = Output_2HIGH_Output1HIGH
error_code = SMifareX1.CMD_WriteOutputPins(Output_Status)
‘Sets the output pins to high
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.14 CMD_WriteE2promKey

This function writes the key to E2prom Block as Master Key.

function TSMifareX.CMD_WriteE2promKey(var E2prom_Block_No, Key_Type: Byte; var Keys: TS_Keys): **Integer**	
Arguments	E2prom_Block_no: Byte Key_Type : Byte Keys : TS_Keys • E2prom_Block_no represents the block number of the eeprom • Key_Type represents the key type of the key • Keys
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim Keys As TS_Keys
Dim error_code As Long
Dim e2prom_block As Byte
Dim Key_Type As Byte

e2prom_block = 0                                    'Block number of e2prom
Key_Type = 0                                        'Key Type is A Type
Keys.Key(0) = &HFF                                  'Key i 'FF FF FF FF FF FF'
Keys.Key(1) = &HFF
Keys.Key(2) = &HFF
Keys.Key(3) = &HFF
Keys.Key(4) = &HFF
Keys.Key(5) = &HFF

error_code = SMifareX1.CMD_WriteE2promKey(e2prom_block, Key_Type, Keys)
'Writes the key to the specified e2prom Block as Master Key
'if error_code is not zero look at the error code table in order to understand
'the error
```

3.15 CMD_ReadBlock

This function reads the specified block of the tag.

function TSMifareX.CMD_ReadBlock(var Block_No: Byte; var Block_Buffer: TS_Mifare_Block; var Block_Str: WideString): **Integer**	
Arguments	Block_No : Byte Block_Buffer : TS_Mifare_Block Block_Str :WideString • Block_No represents the block number of the tag • Block_Buffer is 16 byte of array, values of the Block bytes • Block Str is the characters of the Block bytes
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_No As Byte
Dim my_block As TS_Mifare_Block
Dim my_block_str As String

Block_No = 3
error_code = SMifareX1.CMD_ULReadBlock(Block_No, my_block, my_block_str)
‘reads the third block
‘my_block returns the values of bytes
‘my_block_str is the character-string of the block bytes
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.16 CMD_WriteBlock

This function writes the data to the specified block of the tag.

function TSMifareX.CMD_WriteBlock(var Block_No: Byte; var Block_Buffer: TS_Mifare_Block): **Integer**	
Arguments	Block_No : Byte Block_Buffer : TS_Mifare_Block • Block_No represents the block number of the tag • Block_Buffer is 16 byte of array, hex-values of the Block bytes
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_No As Byte
Dim my_block As TS_Mifare_Block
Dim i As Byte

For i=0 to 15
    my_block.Bytes(i) = &H00
Next i
Block_No = 4

error_code = SMifareX1.CMD_WriteBlock(Block_No, my_block)
‘writes my_block to the fourth block of the tag
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.17 CMD_StrtoBlockBuffer

This function converts the specified string to the 16 byte Block array.

function TSMifareX.CMD_StrtoBlockBuffer(var Content_Str: WideString; var Block_Buffer: TS_Mifare_Block): **Integer**	
Arguments	Content_Str : widestring Block_Buffer : TS_Mifare_Block • Content_str contains characters • Block_Buffer is 16 byte of array, decimal values of the Block bytes
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim My_string As String
Dim Block_Buffer As TS_Mifare_Block

My_string = “SONMICRO MIFARE”
error_code = SMifareX1.CMD_StrtoBlockBuffer(My_string, Block_Buffer)
‘converts the “SONMICRO MIFARE” to type of Mifare_Block
‘Block_Buffer.Bytes(0) → 83
‘Block_Buffer.Bytes(1) → 79 ...
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.18 CMD_BlockBufferToStr

This function converts the specified 16 byte Block array to the character String.

function TSMifareX.CMD_BlockBufferToStr(var Block_Buffer: TS_Mifare_Block; var Block_Str: WideString): **Integer**	
Arguments	Block_Buffer : TS_Mifare_Block Block_Str : widestring • Block_str contains characters • Block_Buffer is 16 byte of array, decimal values of the Block bytes
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim My_string As String
Dim Block_Buffer As TS_Mifare_Block

Block_Buffer.Bytes(0) = 83
Block_Buffer.Bytes(1) = 79
Block_Buffer.Bytes(2) = 78
...

error_code = SMifareX1.CMD_BlockBufferToStr(Block_Buffer, My_string)
‘converts the “SONMICRO MIFARE” to type of Mifare_Block
‘My_String = “SON...”
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.19 CMD_WriteValue

This function writes the value(signed 4 Byte) to the specified block.

function TSMifareX.CMD_WriteValue(var Block_No: Byte; var Block_Value: Integer): **Integer**	
Arguments	Block_No : Byte Block_Value : Integer(4 byte signed) • Block_No represents the number of the block • Block Value is the value of the Block
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_No as Byte
Dim Value As Long

Value = 23400080
Block_No = 5
error_code = SMifareX1.CMD_WriteValue(Block_No, Value)
‘Writes the value to the fifth block
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.20 CMD_ReadValue

This function reads the value(signed 4 Byte) from the specified block.

function TSMifareX.CMD_ReadValue(var Block_No: Byte; var Block_Value: Integer): **Integer**	
Arguments	Block_No : Byte Block_Value : Integer(4 byte signed) • Block_No represents the number of the block • Block Value is the value of the Block
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_No as Byte
Dim Value As Long

Block_No = 5
error_code = SMifareX1.CMD_WriteValue(Block_No, Value)
‘Reads the value to the fifth block, Value variable returns the value
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.21 CMD_IncValue

This function increments the value of the specified block with the specified value.

function TSMifareX.CMD_IncValue(var Block_No: Byte; var Increment_Value, Block_Value: Integer): **Integer**	
Arguments	Block_No : Byte Increment_Value : Integer(4 byte signed) Block_Value : Integer(4 byte signed) • Block_No represents the number of the block • Increment_value represents the value which is added to the block value • Block Value is the value of the Block
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_No as Byte
Dim Inc_value As Long
Dim Value As Long

Block_No = 5
Inc_Value = 3
error_code = SMifareX1.CMD_IncValue(Block_No,Inc_Value, Value)
‘Before increment operation value = 7, after increment operation value is 10
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.22 CMD_DecValue

This function decrements the value of the specified block with the specified value.

function TSMifareX.CMD_DecValue(var Block_No: Byte; var Decrement_Value, Block_Value: Integer)**: Integer**	
Arguments	Block_No : Byte Decrement_Value : Integer(4 byte signed) Block_Value : Integer(4 byte signed) • Block_No represents the number of the block • Increment_value represents the value which is subtracted to the block value • Block Value is the value of the Block
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_No as Byte
Dim Dec_value As Long
Dim Value As Long

Block_No = 5
Dec_Value = 3
error_code = SMifareX1.CMD_IncValue(Block_No,Ddec_Value, Value)
‘Before increment operation value = 7, after increment operation value is 4
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.23 CMD_WriteSectorTrailer

This function writes the bytes to the specified sector trailer block.

function TSMifareX.CMD_WriteSectorTrailer(var Sector_No: Byte; var Access_Bytes: TS_Access_Bytes; var KeyA, KeyB: TS_Keys): **Integer**	
Arguments	Sector_No : Byte AccessBytes : Integer(4 byte signed) KeyA,KeyB : Integer(4 byte signed) • Sector_No represents the number of the sector • Access_Bytes represents the sector trailer access bytes • KeyA, KeyB are the key values
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Sector_No As Byte
Dim Key_A As TS_Keys
Dim Key_B As TS_Keys
Dim my_access As TS_Access_Bytes

Key_A.Key(0) = &HFF
Key_A.Key(1) = &HFF
Key_A.Key(2) = &HFF
Key_A.Key(3) = &HFF
Key_A.Key(4) = &HFF
Key_A.Key(5) = &HFF

Key_B.Key(0) = &HFF
Key_B.Key(1) = &HFF
Key_B.Key(2) = &HFF
Key_B.Key(3) = &HFF
Key_B.Key(4) = &HFF
Key_B.Key(5) = &HFF

my_access.Byte6 = &HFF
my_access.Byte7 = &H07
my_access.Byte8 = &H80
my_access.User_Data = &HEE
Sector_No = 1
error_code = SMifareX1.CMD_WriteSectorTrailer(Sector_No, my_access, Key_A, Key_B)
‘Write these bytes to the sector block
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.24 CMD_CreateAccessBytes

This function creates access bytes with respect to access bits.

function TSMifareX.CMD_CreateAccessBytes(var C0, C1, C2, C3: Byte; var Access_Bytes: TS_Access_Bytes): **Integer**	
Arguments	CO, C1, C2, C3 : Byte AccessBytes : TS_Access_Bytes • C0, C1, C2, C3 represent the block bits. • Access_Bytes represents the sector trailer access bytes
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim my_access As TS_Access_Bytes
Dim Block0 As Byte
Dim Block1 As Byte
Dim Block2 As Byte
Dim Block3 As Byte

Block0 = 0
Block1 = 0
Block2 = 0
Block3 = 4
error_code = SMifareX1.CMD_CreateAccessBytes(Block0, Block1, Block2, Block3,
my_access)
‘my_access.Byte6 = FF
‘my_access.Byte7 = 07
‘my_access.Byte8 = 80
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.25 CMD_ULWriteBlock

This function writes the bytes to the UL tags block.

function TSMifareX.CMD_ULWriteBlock(var Block_No: Byte; var Block_Buffer: TS_Mifare_Block **): Integer**	
Arguments	Block_No : Byte Block_Buffer : TS_Mifare_Block • Block_No represents the block number of the tag • Block_Buffer is 16 byte of array, hex-values of the Block bytes
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Block_Buffer As TS_Mifare_Block
Dim Block_No As Byte

Block_Buffer.Bytes(0) = &HFF
Block_Buffer.Bytes(1) = &HFF
Block_Buffer.Bytes(2) = &HFF
Block_Buffer.Bytes(3) = &HFF
Block_No = 5

error_code = SMifareX1.CMD_ULWriteBlock(Block_No, Block_Buffer)
‘write Block_buffer to the fifth block
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.26 CMD_ULWriteOtp

This function writes the bytes to the OTP block of the UL tag.

function TSMifareX.CMD_WriteOtp(var OTP_Bytes: TS_UL_Otp): **Integer**	
Arguments	OTP_Bytes : TS_UL_Otp • New_Baud represents the new buad rate of the device
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim OTP As TS_UL_Otp

OTP.Bytes(0) = &H00
OTP.Bytes(1) = &H00
OTP.Bytes(2) = &H00
OTP.Bytes(3) = &H00

error_code = SMifareX1.CMD_ULWriteOtp(OTP)
‘Write 00 to all OTP bytes
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.27 CMD_ULWriteLock

This function writes the Lock bytes to the Lock block of the UL tag.

function TSMifareX.CMD_WriteLock(var Lock_Bytes: TS_UL_Lock): **Integer**	
Arguments	Lock_Bytes: TS_UL_Lock • Lock bytes represents the bytes of the Lock block
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Lock_Byte As TS_UL_Lock
Dim sonuc As Integer

Lock_Byte.Bytes(0) = &H00
Lock_Byte.Bytes(1) = &H00
error_code = SMifareX1.CMD_ULWriteLock(Lock_Byte)
'Write 00 to all Lock  bytes
'if error_code is not zero look at the error code table in order to understand
'the error
```

3.28 CMD_SetDeviceBaudRate

This function sets the device baud rate to the specified value.

function TSMifareX.CMD_SetDeviceBaudRate(var New_Baud: Integer): **Integer**	
Arguments	New_Baud : Integer • New_Baud represents the new buad rate of the device
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long
Dim Baud_rate As Long

Baud_rate = 19200
error_code = SMifareX1.CMD_SetDeviceBaudRate(Baud_rate)
‘Changes the baud rate to 19200
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.29 CMD_Sleep

This function takes the module to Sleep.

function TSMifareX.CMD_Sleep( ): **Integer**	
Arguments	No Arguments
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long

error_code = SMifareX1.Sleep()
‘Takes the module to sleep
‘if error_code is not zero look at the error code table in order to understand
‘the error
```

3.30 CMD_SwitchOnRf

This function takes RF power on.

function TSMifareX.CMD_SwitchOnRf( ): **Integer**	
Arguments	No Arguments
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long

error_code = SMifareX1.SwitchOnRf ()
'Takes the Antenna Power on
'if error_code is not zero look at the error code table in order to understand
'the error
```

3.31 CMD_SwitchOffRf

This function takes RF power off.

function TSMifareX.CMD_SwitchOffRf( ): **Integer**	
Arguments	No Arguments
Returns	0 : Successful Others : Look at the error code table

Example for VB:

```
Dim error_code As Long

error_code = SMifareX1.SwitchOffRf ()
'Takes the Antenna Power off
'if error_code is not zero look at the error code table in order to understand
'the error
```

3.32 CMD_OnReceive

This procedure is used to get the data in SMifareX1_OnPacketReceive event.

procedure TSMifareX.CMD_OnReceive(var Data_Type: Byte; var Packet_Str: WideString)	
Arguments	Data Type : Byte Packet_Str : Widestring • Data Type represents the type of the incoming packet • Packet_Str represents the incoming data packet

Example for VB:

```
Dim tag_type As Byte
Dim my_packet As String

Call SMifareX1.CmdOnReceive(tag_type, my_packet)

‘if tag_type = 2 and my_packet C7890654 → 1K Tag Received nad Serial is 540689C7
```

4. ERROR CODE TABLE

Error Codes	
Codes	Meanings
0	Process is done successfully
1	Timeout
2	Unexpected response
3	Data greater than 64 byte
4	Checksum error
5	RF Field is OFF
6	No Tag or Login Failed
7	Argument out of range or Wrong Command
8	Invalid Key Format
9	Read Failed
10	Unable to read after write
11	Write Failed
12	Invalid Value Block
13	Change of Baud rate Failed
14	Com Port has not been created yet
15	Sector trailer block, use sector write function
16	Serial Block, you can not write anything
17	OTP or Lock Block
18	Com port is already open
19	Com port could not be created

5. SALES AND SERVICE INFORMATION

To obtain information about SonMicro Electronics products and technical support, reference the following information.

SonMicro ELECTRONICS LTD.
Cankaya M. Soguksu C.
Aslihan Ishani 2/15
Mersin, 33070
TURKIYE

Phone: +90 324 237 21 28
Facsimile: +90 324 237 21 86
Email: info@sonmicro.com
Web Site: http://www.sonmicro.com

Sales	http://www.sonmicro.com/sales.php
Support	http://www.sonmicro.com/ask.php
Documents & Software	http://www.sonmicro.com/1356/d1356.php
User Forums	http://www.sonmicro.com/forums/