
YOWO RFID Module SDK API

For Wince

User Manual V1.0

YOWO RFID Technology Co.,LTD

Content

1	Summary	4
2	HF RFID Module Models	4
3	RFID Module API For Wince	5
3.1.1	DES Encryption or Decryption	5
3.1.2	3DES Encryption or Decryption	5
3.1.3	3DES Encryption or Decryption with vector	6
3.1.4	Open COM Port	6
3.1.5	Close COM Port	7
3.1.6	Change Module UART Baud	7
3.1.7	Open or Close RF Antenna	7
3.1.8	Set RFID Module Request Card Type	8
3.1.9	Write Module Flash Memory	8
3.1.10	Read Module Flash Memory	8
3.1.11	Module Sleep, Any Command Can active it.	9
3.1.12	Get IIC Address	9
3.1.13	Read Module Flash Memory	错误!未定义书签。
3.1.14	Set Module IO Value.....	9
3.1.15	Set Module Works Active Mode or not.....	10
3.1.16	Set Module Works Active Mode or not.....	错误!未定义书签。
3.1.17	Read a block of M1 Card	10
3.1.18	Write a block of M1 Card	11
3.1.19	Read a Sector of M1 Card.....	12
3.1.20	Initial block to a value	12
3.1.21	Read the value(Banlance).....	13
3.1.22	Inc the value	13
3.1.23	Inc the value	14
3.1.24	Backup the value	15
3.1.25	Card Halt.....	15
3.1.26	Backup the value	错误!未定义书签。
3.1.27	ISO14443A CPU Card Reset	16
3.1.28	ISO14443A CPU Card Execute COS.....	16
3.1.29	Read UltraLight Card Data	17
3.1.30	Write UltraLight Card Data	17
3.1.31	ISO14443A CPU Card Reset(Only for YW-401)	17
3.1.32	ISO14443A CPU Card Execute COS(Only for YW-401)	18
3.1.33	ISO14443B CPU Card Reset	18
3.1.34	ISO14443B CPU Card Execute COS	18
3.1.35	ISO14443B CPU Card Halt	19
3.1.36	SR512 Card Active	19
3.1.37	SR512 Card Get UID.....	19
3.1.38	SR512 Card Reset To Inventory.....	20
3.1.39	Read SR512 Card Data	20
3.1.40	Write SR512 Card Data.....	20

3.1.41	SR512 Completion.....	21
3.1.42	ISO15693 Card Inventory	21
3.1.43	ISO15693 Card Stay Quiet	21
3.1.44	ISO15693 Card Select	21
3.1.45	ISO15693 Card Reset to Ready	22
3.1.46	ISO15693 Card Read Block	22
3.1.47	ISO15693 Card Write Block	23
3.1.48	ISO15693 Card Lock Block.....	23
3.1.49	ISO15693 Card Write AFI.....	24
3.1.50	ISO15693 Card Write DSFID	24
3.1.51	ISO15693 Card Lock DSFID	25
3.1.52	ISO15693 Card Get System Information.....	25
4	Order Information	26

1 Summary

YOWO RFID HF RFID Modules are based on high integrated reader ICs. YOWO RFID module has many products for more requirement. They can driver by ARM or SCM, it can receive command from uart or iic. Also they can placed in a device which run with Wince os. This File descript how to program on Wince with C#.

The RFID module support ISO14443-A Mifare One S50,S70, UltraLight, MifarePro, ISO14443-B SR176, ISO15693 ,I CODE SL2 and other compatible card.

2 HF RFID Module Models

Type	Card Protocol	Vcc	Interface	Size (mm)	Support RFID Card
YW-201	ISO14443A	+5V	IIC,UART(TTL)	20*40	Mifare S50,S70,CPU
YW-201C	ISO14443A	+5V	IIC,UART(TTL)	60*60	Mifare S50,S70,CPU
YW-201C3	ISO14443A	+3.3V	IIC,UART(TTL)	60*60	Mifare S50,S70,CPU
YW-202	ISO14443A ISO14443B	+5V	IIC,UART(TTL)	20*40	Mifare S50,S70,CPU,SR512
YW-202C	ISO14443A ISO14443B	+5V	IIC,UART(TTL)	60*60	Mifare S50,S70,CPU,SR512
YW-203	ISO15693	+5V	IIC,UART(TTL)	20*40	ICode,TagIT,, MB89R118B
YW-203C	ISO15693	+5V	IIC,UART(TTL)	60*60	ICode,TagIT,, MB89R118B
YW-204	ISO14443A ISO14443B ISO15693	+5V	IIC,UART(TTL)	20*40	Mifare S50,S70,CPU,SR512, ICode,TagIT,, MB89R118B
YW-204C	ISO14443A ISO14443B ISO15693	+5V	IIC,UART(TTL)	60*60	Mifare S50,S70,CPU,SR512, ICode,TagIT,, MB89R118B

YW-401	ISO14443A	+3.3V	IIC,UART(TTL)	38*38	Mifare S50,S70,CPU
YW-411	ISO14443A	+3.3V	UART(TTL)	45*45	Mifare S50,S70

3 RFID Module API For Wince

3.1.1 DES Encryption or Decryption

Prototype: `public static extern int DES(byte cModel, byte[] pkey, byte[] pInData, byte[] pOutData);`

Parameter List:

PARAM	Type	DIR	Implication
cModel	byte	IN	DES mode: 0x00->Encryption 0x01->Decryption
pkey	byte[]	IN	DES Key(8 bytes)
pInData	byte[]	IN	Raw data(8 bytes)
pOutData	byte[]	OUT	Encrypted data(8 bytes)

Return: 0x01->successful, <=0 fail

3.1.2 3DES Encryption or Decryption

Prototype: `public static extern int DES3(byte cModel, byte[] pKey, byte[] pInData, byte[] pOutData);`

Parameter List:

PARAM	Type	DIR	Implication
cModel	byte	IN	DES mode: 0x00->Encryption 0x01->Decryption

pkey	byte[]	IN	DES Key(16 bytes)
pInData	byte[]	IN	Raw data(8 bytes)
pOutData	byte[]	OUT	Encrypted data(8 bytes)

Return: 0x01-> successful, <=0 fail

3.1.3 3DES Encryption or Decryption with vector

Prototype: `public static extern int DES3_CBC(byte cModel, byte[] pKey, byte[] pInData, byte[] pOutData, byte[] pIV);`

Parameter List:

PARAM	Type	DIR	Implication
cModel	byte	IN	DES mode: 0x00->Encryption 0x01->Decryption
pkey	byte[]	IN	DES Key(16 bytes)
pInData	byte[]	IN	Raw data(8 bytes)
pOutData	byte[]	OUT	Encrypted data(8 bytes)
pIV	byte[]	IN	Vector(8 bytes)

Return: 0x01-> successful, <=0 fail

3.1.4 Open COM Port

Prototype: `public static extern int YW_ComInitial(int PortIndex, int Baud)`

Parameter List:

PARAM	Type	DIR	Implication
PortIndex	int	IN	COM Port Index
Baud	int	IN	Baud

Return: 0x01-> successful, <=0 fail

3.1.5 Close COM Port

Prototype: `public static extern int YW_ComFree()`

Parameter List: None

Return: 0x01-> successful, <=0 fail

3.1.6 Change Module UART Baud

Prototype: `public static extern int YW_ComNewBaud(int NewBaudIndex)`

Parameter List:

PARAM	Type	DIR	Implication
NewBaudIndex	int	IN	0x00->9600 0x01->19200 0x02->38400 0x03->57600 0x04->115200

Return: 0x01-> successful, <=0 fail

3.1.7 Open or Close RF Antenna

Prototype: `public static extern int YW_AntennaStatus(bool AntOpen, bool AutoRequest)`

Parameter List:

PARAM	Type	DIR	Implication
AntOpen	bool	IN	Open Antenna or Close True: Open False: Close;
AutoRequest	bool	IN	Auto Request or not, if control by wince, it is false

Return: 0x01-> successful, <=0 fail

3.1.8 Set RFID Module Request Card Type.

Prototype: `public static extern int YW_SearchCardMode(byte Mode)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	0x41 - type A(0x41) 0x42 - typeB(0x42) 0x31 - IS015693(0x31) 0x53 - SR512

Return: 0x01-> successful, <=0 fail

3.1.9 Write Module Flash Memory

Prototype: `public static extern int YW_WriteReaderMemory(int AddrBegin, int DataLen, byte[] Data)`

Parameter List:

PARAM	Type	DIR	Implication
AddrBegin	int	IN	Begin Addr
DataLen	int	IN	Bytes Nums want to write
Data	byte[]	IN	Data

Return: 0x01-> successful, <=0 fail

3.1.10 Read Module Flash Memory

Prototype: `public static extern int YW_ReadReaderMemory(int AddrBegin, int DataLen, byte[] DataRead)`

Parameter List:

PARAM	Type	DIR	Implication
AddrBegin	int	IN	Begin Addr

DataLen	int	IN	Bytes Nums want to write
Data	byte[]	OUT	Data

Return: 0x01-> successful, <=0 fail

3.1.11 Module Sleep, Any Command Can active it.

Prototype: public static extern int YW_ModuleSleep()

Parameter List: None

Return: 0x01-> successful, <=0 fail

3.1.12 Get IIC Address

Prototype: public static extern int YW_GetIICAddr ()

Parameter List: None

Return: <=0 fail, >0 IIC address

3.1.13 Set IIC Address

Prototype: public static extern int YW_SetIICAddr(byte IICAddr)

Parameter List:

PARAM	Type	DIR	Implication
IICAddr	byte	IN	New IIC Address

Return: 0x01-> successful, <=0 fail

3.1.14 Set Module IO Value

Prototype: public static extern int YW_SetIOValue(byte IOindex ,byte IOValue)

Parameter List:

PARAM	Type	DIR	Implication
-------	------	-----	-------------

IOindex	byte	IN	Must be 0
IOValue	byte	IN	0 or 1

Return: 0x01-> successful, <=0 fail

3.1.15 Set Module Works Active Mode or not

Prototype: `public static extern int YW_SetAutoRequestMode(bool bAutoMode);`

Parameter List:

PARAM	Type	DIR	Implication
bAutoMode	bool	IN	True: Active Mode False: Command Mode

Return: 0x01-> successful, <=0 fail

3.1.16 ISO14443A Request Card

Prototype: `public static extern int YW_RequestCard(byte CardMode, byte[] CardNo, byte[] OtherInfo)`

Parameter List:

PARAM	Type	DIR	Implication
CardMode	byte	IN	0x00: request all card in antenna area 0x01: request card that have not halted in
CardNo	byte[]	OUT	Card No.
OtherInfo	byte[]	OUT	Other Infomation

Return: <=0 fail, else return value(>0) is the length of CardNo.

3.1.17 Read a block of M1 Card

Prototype: `public static extern int YW_ReadaBlock(byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int BlockAddr, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
BlockAddr	int	IN	Block Index
pData	byte[]	OUT	16 bytes Data

Return: 0x01-> successful, <=0 fail

3.1.18 Write a block of M1 Card

Prototype: `public static extern int YW_WriteaBlock (byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int BlockAddr, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
BlockAddr	int	IN	Block Index
pData	byte[]	IN	16 bytes Data

Return: 0x01-> successful, <=0 fail

3.1.19 Read a Sector of M1 Card

Prototype: `public static extern int YW_ReadSector (byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int SectorAddr, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
SectorAddr	int	IN	Sector Index
pData	byte[]	OUT	64 bytes Data

Return: 0x01-> successful, <=0 fail

3.1.20 Initial block to a value

Prototype: `public static extern int YW_InitiaPurse (byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int BlockAddr, int IniMoney)`

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key

KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
BlockAddr	int	IN	Block Index
IniMoney	int	IN	Initial Value

Return: 0x01-> successful, <=0 fail

3.1.21 Read the value(Banlance)

Prototype: public static extern int YW_ReadPurse (byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int BlockAddr, ref int Money)

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
BlockAddr	int	IN	Block Index
Money	Ref int	OUT	Initial Value

Return: 0x01-> successful, <=0 fail

3.1.22 Inc the value

Prototype: public static extern int YW_IncPurse((byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int BlockAddr, int Money)

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
BlockAddr	int	IN	Block Index
Money	int	IN	Inc Value

Return: 0x01-> successful, <=0 fail

3.1.23 Dec the value

Prototype: `public static extern int YW_DecPurse((byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int BlockAddr, int Money)`

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
BlockAddr	int	IN	Block Index
Money	int	IN	Dec Value

Return: 0x01-> successful, <=0 fail

3.1.24 Backup the value

Prototype: `public static extern int YW_BakPurse(byte KeyAB, byte KeyMode, byte KeyIndx, byte[] Key, int SrcBlockAddr, int DesBlockAddr)`

Parameter List:

PARAM	Type	DIR	Implication
KeyAB	byte	IN	0x00: Key A 0x01: Key B
KeyMode	byte	IN	0x00: Command Key 0x01: System Key
KeyIndx	byte	IN	0 to 31 if KeyMode =1(System Key), KeyIndx=0 if KeyMode =0(Command Key),
Key	byte[]	IN	6 bytes.
SrcBlockAddr	int	IN	Source block Index
DesBlockAddr	int	IN	Des block Index SrcBlockAddr and DesBlockAddr must be the same sector

Return: 0x01-> successful, <=0 fail

3.1.25 Card Halt

Prototype: `public static extern int YW_Halt_()`

Parameter List: None

Return: 0x01-> successful, <=0 fail

3.1.26 Down System Key

Prototype: `public static extern int YW_DownKey(int KeyIndex, byte[] Key)`

Parameter List:

PARAM	Type	DIR	Implication
KeyIndex	byte	IN	0 to 31, System Key Index
Key	byte[]	IN	6 Bytes, System Key

Return: 0x01-> successful, <=0 fail

3.1.27 ISO14443A CPU Card Reset

Prototype: `public static extern int YW_ACPU_Reset(ref int rtLen, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
rtLen	ref int	IN	Length of pData
pData	byte[]	OUT	Reset data

Return: 0x01-> successful, <=0 fail

3.1.28 ISO14443A CPU Card Execute COS

Prototype: `public static extern int YW_ACPU_COS(int LenCOS, byte[] Com_COS, ref int rtLen, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
LenCOS	int	IN	Length of COS
Com_COS	byte[]	IN	COS
rtLen	ref int	OUT	Length of pData
pData	byte[]	OUT	Data after execute cos

Return: 0x01-> successful, <=0 fail

3.1.29 Read UltraLight Card Data

Prototype: `public static extern int YW_M401UL_Read(byte Block, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
Block	byte	IN	Block Index
pData	byte[]	OUT	Block Data

Return: 0x01-> successful, <=0 fail

3.1.30 Write UltraLight Card Data

Prototype: `public static extern int YW_M401UL_Write (byte Block, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
Block	byte	IN	Block Index
pData	byte[]	IN	Block Data

Return: 0x01-> successful, <=0 fail

3.1.31 ISO14443A CPU Card Reset(Only for YW-401)

Prototype: `public static extern int YW_M401ACPU_Reset(ref int rtLen, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
rtLen	ref int	IN	Length of pData
pData	byte[]	OUT	Reset data

Return: 0x01-> successful, <=0 fail

3.1.32 ISO14443A CPU Card Execute COS(Only for YW-401)

Prototype: `public static extern int YW_ACPU_COS(int LenCOS, byte[] Com_COS, ref int rtLen, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
LenCOS	<code>int</code>	IN	Length of COS
Com_COS	<code>byte[]</code>	IN	COS
rtLen	<code>ref int</code>	OUT	Length of pData
pData	<code>byte[]</code>	OUT	Data after execute cos

Return: 0x01-> successful, <=0 fail

3.1.33 ISO14443B CPU Card Reset

Prototype: `public static extern int YW_BCPU_Reset(byte Mode, ref int rtLen, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	<code>byte</code>	IN	0x00: Active Card 0x01: All Card
rtLen	<code>ref int</code>	OUT	Length of pData
pData	<code>byte[]</code>	OUT	Reset data

Return: 0x01-> successful, <=0 fail

3.1.34 ISO14443B CPU Card Execute COS

Prototype: `public static extern int YW_BCPU_COS(int LenCOS, byte[] Com_COS, ref int rtLen, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
LenCOS	int	IN	Length of COS
Com_COS	byte[]	IN	COS
rtLen	ref int	OUT	Length of pData
pData	byte[]	OUT	Data after execute cos

Return: 0x01-> successful, <=0 fail

3.1.35 ISO14443B CPU Card Halt

Prototype: public static extern int YW_BCPU_Halt(byte[] PUPI)

Parameter List:

PARAM	Type	DIR	Implication
PUPI	byte[]	IN	4bytes, PUPI

Return: 0x01-> successful, <=0 fail

3.1.36 SR512 Card Active

Prototype: public static extern int YW_SR512_Active(byte[] CHIPID)

Parameter List:

PARAM	Type	DIR	Implication
CHIPID	byte[]	OUT	1 byte, Discard

Return: 0x01-> successful, <=0 fail

3.1.37 SR512 Card Get UID

Prototype: public static extern int YW_SR512_GetUID(byte[] UID)

Parameter List:

PARAM	Type	DIR	Implication
-------	------	-----	-------------

UID	byte[]	OUT	8 bytes, UID of SR512
-----	--------	-----	-----------------------

Return: 0x01-> successful, <=0 fail

3.1.38 SR512 Card Reset To Inventory

Prototype: `public static extern int YW_SR512_ResetToInventory()`

Parameter List: None

Return: 0x01-> successful, <=0 fail

3.1.39 Read SR512 Card Data

Prototype: `public static extern int YW_SR512_Read(int BlockID, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
BlockID	int	IN	Block Index
pData	byte[]	OUT	Block Data

Return: 0x01-> successful, <=0 fail

3.1.40 Write SR512 Card Data

Prototype: `public static extern int YW_SR512_Write(int BlockID, byte[] pData)`

Parameter List:

PARAM	Type	DIR	Implication
BlockID	byte	IN	Block Index
pData	byte[]	IN	Block Data

Return: 0x01-> successful, <=0 fail

3.1.41 SR512 Completion

Prototype: YW_SR512_Completion()

Parameter List: None

Return: 0x01-> successful, <=0 fail

3.1.42 ISO15693 Card Inventory

Prototype: public static extern int YW_ISO15693_Inventory(byte[] PUID, ref byte DSFID, ref byte PLen)

Parameter List:

PARAM	Type	DIR	Implication
PUID	byte[]	OUT	UID of ISO15693
DSFID	ref byte	OUT	DSFID
PLen	ref byte	OUT	The Length of UID

Return: 0x01-> successful, <=0 fail

3.1.43 ISO15693 Card Stay Quiet

Prototype: public static extern int YW_ISO15693_Stay_Quiet(byte[] PUID)

Parameter List:

PARAM	Type	DIR	Implication
PData	byte[]	IN	UID of ISO15693

Return: 0x01-> successful, <=0 fail

3.1.44 ISO15693 Card Select

Prototype: public static extern int YW_ISO15693_Select (byte[] PUID)

Parameter List:

PARAM	Type	DIR	Implication
PData	byte[]	IN	UID of ISO15693

Return: 0x01-> successful, <=0 fail

3.1.45 ISO15693 Card Reset to Ready

Prototype: `public static extern int YW_ISO15693_Reset_To_Ready(byte Mode, byte[] PUID)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693

Return: 0x01-> successful, <=0 fail

3.1.46 ISO15693 Card Read Block

Prototype: `public static extern int YW_ISO15693_Read(byte Model, byte[] PUID, byte StartBlockID, byte BlockNums, byte[] PData, ref byte PLen)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before

PUID	byte[]	IN	UID of ISO15693
StartBlockID	byte	IN	Begin block Index you read
BlockNums	byte	IN	The block numbers you read
PData	Byte[]	OUT	Data read
PLen	ref byte	OUT	The length of data

Return: 0x01-> successful, <=0 fail

3.1.47 ISO15693 Card Write Block

Prototype: `public static extern int YW_ISO15693_Write(byte Model, byte[] PUID, byte BlockID, byte DataLen, byte[] PData)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693
BlockID	byte	IN	block Index you write
DataLen	byte	IN	The length of data
PData	Byte[]	IN	Data

Return: 0x01-> successful, <=0 fail

3.1.48 ISO15693 Card Lock Block

Prototype: `public static extern int YW_ISO15693_Lock_Block(byte Model, byte[] PUID, byte BlockID)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693
BlockID	byte	IN	block Index you lock

Return: 0x01-> successful, <=0 fail

3.1.49 ISO15693 Card Write AFI

Prototype: `public static extern int YW_ISO15693_Write_AFI(byte Model, byte[] PUID, byte AFI)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693
AFI	byte	IN	AFI

Return: 0x01-> successful, <=0 fail

3.1.50 ISO15693 Card Write DSFID

Prototype: `public static extern int YW_ISO15693_Write_DSFID(byte Model, byte[] PUID, byte DSFID)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693
DSFID	byte	IN	DSFID

Return: 0x01-> successful, <=0 fail

3.1.51 ISO15693 Card Lock DSFID

Prototype: `public static extern int YW_ISO15693_Lock_DSFID(byte Model, byte[] PUID)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693

Return: 0x01-> successful, <=0 fail

3.1.52 ISO15693 Card Get System Information

Prototype: `public static extern int YW_ISO15693_Get_System_Information(byte Model, byte[] PUID, byte[] PData, ref byte PLen)`

Parameter List:

PARAM	Type	DIR	Implication
Mode	byte	IN	Mode=0, the param PUID can be discarded, but we must excute select Command before . Mode=2, the param PUID is the Card UID which you want, you need not excute select Command before
PUID	byte[]	IN	UID of ISO15693
PData	byte[]	OUT	The data of system information
PLen	ref byte	OUT	The length of data

Return: 0x01-> successful, <=0 fail

4 Order Information

Company: YOWO RFID Technology Co., LTD

Web: <http://www.yoworfid.com>

<http://www.youwokeji.com.cn>

Mail: coodor@126.com