915MHz Standard Reader
ADAcitveX Use Guide

Menu

1. Function library description

The dynamic library has 3 files, including ADActiveX.dll, ADDevice.dll, ADDeviceReader.dll。

The library is only suitable for the reader working in passive mode or response mode; when the reader works in active mode, the library may cause data conflicts or overlap errors.

2. Device Interface

Suitable for adoption RS-232 ,RS-485,TCPIP,HID to connect reader.

3. Variable

Variable Name	Type	Description
IsConnected	bool	Get Connection status
FCount	int	Return status code
RecvBytes	List<byte>	Receive data byte arrays
RecvString	string	Receive data hex string
Address	int	Reader's address for data transmission
Version	string	Reader's firmware version
ReaderMode	string	Reader's type
DelayTime	int	Communication timeout time, unit 10ms, default value 20*10ms
Type	int	Reader’s type, 0 - 915MHz Standard version 1 - 915MHz Phychips version 2 - 2.4GHz Standard version
Mode	int	Reader’s Comm Type 0 - RS232/RS485, 1 - TCPIP/WIFI, 2 - HID

4. Public Function

4.1. GetReturnCode

```
string GetReturnCode(int state)          // Get return code description (English)
string GetReturnCode()                   // Get return code description (English)
```

5. Funcation

5.1. Device Management Funcation

5.1.1. Connect

```
virtual bool Connect(string hostOrCom, int baudOrPort, int type = 0)
```

Function: Create a communication connection
Parameters: hostOrCom：Reader’s IP Address or Serial port Name
             baudOrPort：Reader’s IP Port or Serial port baudrate
             type：Reader’s Type(def 0,[0~2])
Return: Return Code，See return code table

Example:

```
        Connect("COM1", 9600);                  // Serial
        Connect("192.168.2.115", 49152);        //Net
        Connect("AD", 0);                       //HID
```

5.1.2. DisConnect

```
void DisConnect()
```

Function: Disconnect communication
Parameters: Non
Return: Non

Example: `DisConnect();`

5.1.3. Information

```
int Information()
int Information(ref int oAddress, ref string oVer, ref string oMode)
```

Function:	Get device address and version information
Parameters:	Non

Return:	Return Code，See return code table

5.1.4.pSetAddress

int pSetAddress(int iData)

Function:	Set up device address
Parameters:	iData：	(1~65535)

Return:	Return Code，See return code table

5.1.5.pGetConfig

int **pGetConfig**(ref byte[] oData)

Function:	Get basic parameters of device
Parameters:	oData：Returns the base parameter byte array. (See basic parameter table)

Return:	Return Code，See return code table

5.1.6.pSetConfig

int pSetConfig(byte[] iData)

Function:	Set basic parameters of device
Parameters:	iData：Byte Arrays (See basic parameter table)

Return:	Return Code，See return code table

5.1.7.pGetTcpip

Non standard commands are outdated;
int pGetTcpip(ref byte[] oData)

Function:	Get device network communication parameters
Parameters:	oData：	TCPIP parameters Byte Arrays. (See TCPIP Parameter Table)

Return:	Return Code，See return code table

5.1.8.pSetTcpip

Non standard commands are outdated;

int pSetTcpip(byte[] iData)

Function: Set device network communication parameters
Parameters: iData：TCPIP parameters byte arrays. (See TCPIP Parameter Table)

Return: Return Code，See return code table

5.2. ISO18000-6B Function

5.2.1.pIdentify6B

int pIdentify6B(ref byte[] oData)

Function: Identify card from ISO18000-6B tag
Parameters: oData：Byte array contains antenna numbers.

Return: Return Code，See return code table

Example:

```
int   oData[13];
int   st;
st = pIdentify6B( ref oData);
```

Notes: Get the unique ID number in the tag, get 13 bytes of data, the first byte is the antenna number, generally 2-9 bytes are the card number, and the last four bytes default to 0;

5.2.2.pRead6B

int pRead6B(int iStart, int iLength, ref byte[] oData)

Function: Read data from ISO18000-6B tag
Parameters: iStart：Start address (byte)
iLength：Data length (byte)
oData：Return data array contains antenna numbers

Return: Return Code，See return code table

Example:

```
int   oData[13];
int   iAddr = 18;
int   iSize = 12;
int   st;
st = pRead6B(iAddr, iSize, ref oData);
```

5.2.3.pWrite6B

```
int pWrite6B(int iStart, int iLength, byte[] iData)
```

Function: Write Data to ISO18000-6B tag

Parameters: iStart：Start address (byte)
iLength：Data length (byte)
iData：Data (byte array)

Return: Return Code，See return code table

Example:

```
int   iData [2]={0x01,0x02};
int   iAddr = 18;
int   iSize = 2;
int   st;
st = pWrite6B(iAddr, iSize, iData);
```

Notes: Write data to the label, the address starts from 18, and the front location data is non modifiable.

5.3. EPC(GEN 2) Function

5.3.1.pIdentify6C

```
int pIdentify6C(ref byte[] oData)
```

Function: Identify card from ISO18000-6C(EPC) tag

Parameters: oData：Return data array contains antenna numbers

Return: Return Code，See return code table

Example:

```
int   oData[13];
int   st;
st = pIdentify6C(ref oData);
```

Notes: Get the 12 byte data in the EPC area of the tag.

5.3.2.pRead6C

Double byte operation;

```
int pRead6C(int iMem, int iStartWord, int iLengthWord, ref byte[] oData)
```

Function: Read data from ISO18000-6C(EPC) tag

Parameters: iMem：Memory Bank, 0x00-RFU,0x01-EPC,0x02-TID,0x03-User
iStartWord：Start Address(word)
iLengthWord：Data Length in Words(word)
oData：Return data array contains antenna numbers

Return: Return Code，See return code table

Example:

```
int   oData[13];
int   iMem = 1;
int   iAddr = 2;
int   iSize = 2;
int st;
st = pRead6C(iMem, iAddr, iSize, ref oData);
```

Tag storage partition:

Block	Memory Contents	Block address	Capacity bytes	Capacity bits	Read / write
Reserved	KILL PASSWORD and ACCESS PASSWORD	00H	8	64bits	Readable Writable
EPC	EPC Number	01H	12	96bits	Readable Writable
TID	identification number, Each TID number should be unique.	02H	24	196bits	Read-only
USER	User defined data	03H	64	512bits	Readable Writable

Notes: **The EPC area data address starts at 2,**
In each area, each address stores 2 bytes.

Example:

EPC Number: 01 02 03 04 05 06 07 08 09 10 11 12;

iMem = 1; iAddr = 2; iSize = 4;
Get data: 01 01 02 03 04 (The front 01 is the antenna number, the integrated reader is 01, and the multichannel reader is the channel number.)

iMem = 1; iAddr = 3; iSize = 4;
Get data: 01 03 04 05 06 ;

iMem = 1; iAddr = 4; iSize = 4;
Get data: 01 05 06 07 08 ;

Analogy…

5.3.3.pWrite6C

Double byte operation;

int pWrite6C(int iMem, int iStartWord, int iLengthWord, byte[] iData)

Function: Write Data to ISO18000-6C(EPC) tag

Parameters: iMem：Memory Bank, 0x00-RFU,0x01-EPC,0x02-TID,0x03-User
iStartWord：Start Address(word)
iLengthWord：Data Length in Words(word) [No more than 16byte(8Words) one operation.]
iData：data(byte array)

Return: Return Code，See return code table

Example:

```
int   iData [2]={0x01,0x02};
int   iMem = 1;
int   iAddr = 2;
int   iSize = 2;
int   st;
st = pWrite6B(iMem, iAddr, iSize, iData);
```

6. Return Code Table

Code	Status	Code	Status	Code	Status
0	Succeed	1	Fail	10	Succeed Non Data
100	Card Error	101	Action Error(get or set)		
201	Connect Error	211	Connect USB Error	252	Address error
202	Get Error	212	Enter Program error	253	Protocol error
203	Set Error	213	Exit Program error		
204	Set Timeout				
205	Send Error				
206	Receive Error				
207	Disconnect Error				
208	Send Timeout				
209	Receive Timeout				

7. Appendix Parameter Table

7.1. Base Parameter Table

Parameter	Description	Reference Value
Para1	Power Size	Adjustable reader to read tag distance(Nonlinearity) Default:30 Reference:(decimal format) 0~30
Para2	Hopping Enable	Fixed frequency or frequency hopping mode Default:1 Reference:(decimal format) 1- Fixed 2- Hopping

Para3	Fixed Freq	Default:110(915MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para4	Hopping1	Default:84(902MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para5	Hopping2	Default:93(906.5MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para6	Hopping3	Default:102(911MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para7	Hopping4	Default:110(915MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para8	Hopping5	Default:119(919.5MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para9	Hopping6	Default:130(925MHz)
		Reference:(decimal format)
		0~200(860MHz ~ 960MHz)

Para10	Work Mode	Command： The reader stops working, wait commands and the reader works and acts according to the instruction. Active： The reader working, when read tag then send data to terminal. Passive： The reader working, when read tag then save last tag data in reader buffer. Don’t send data to terminal. Default:2 Reference:(decimal format) 1- Command 2- Active 3- Passive

Para11	Read Interval	Default:10(*1ms)
		Reference:(decimal format)
		5~255(* 1ms)

Para12	Trigger	Default:0 Reference:(decimal format) 0- Close 1- 2-Lower
Para13	Output Mode	Work Mode set to Active, then data auto send to for this interface. Default:1 Reference:(decimal format) 1- RS232 2- RS485 3- TCPIP 4- CANBUS 5- SYRIS 6- Wiegand26 7- Wiegand34
Para14	WG Para 1 – Byte Offset	Reference Wiegand Controller Default:0 Reference:(decimal format) 0~20

Para15	WG Para 2 – Out Interval	Reference Wiegand Controller
		Default:30(* 10ms)
		Reference:(decimal format)
		0~255(* 10ms)

Para16	WG Para 3 – Pulse Width	Reference Wiegand Controller Default:10(* 10us) Reference:0~255(* 10us)

Para17	WG Para 4 – Pulse Period	Reference Wiegand Controller
		Default:15(* 100us)
		Reference:(decimal format)
		0~255(* 100us)

Para18	Antenna	One byte data, low 4 bits for 4 antennas. Example: Antenna 1: 01H(Binary 0000 0001)

		Antenna 3: 04H(Binary 0000 0100) Antenna 1 and Antenna 3: 05H(Binary 0000 0101)
Para19	Tag Type	Default:16 Reference:(decimal format) 1-ISO18000-6B Single tag 16-EPC(GEN 2) Single tag 17-EPC(GEN 2) + ISO18000-6B 32-EPC(GEN 2) Mult tag 64-EPC(GEN 2)+Other Data
Para20	Same ID interval	Default:1s Reference:(decimal format) 0~255s
Para21	Buzzer	Default:1 Reference:(decimal format) 0- Disenabled 1- Enabled
Para22	Other data Area	Tag Type = EPC(GEN 2)+Other Data, This parameters is other block choose. Default:1 Reference:(decimal format) 1-TID 2-USER
Para23	Other data Start Address	Tag Type = EPC(GEN 2)+Other Data, This parameters is other block start address. Default:0 Reference:(decimal format) 0~31
Para24	Other data Length	Tag Type = EPC(GEN 2)+Other Data, This parameters is other block data length. Default:2 Reference:(decimal format)1~12
Para25	Encrypt	Using encryption function； Default: 0 Reference:(decimal format) 0- General version, not encrypted； 1- Use reader encryption；
Para26	Password	Default:0000 Reference:(decimal format) 0000~9999 Example: password 0123(Dec) = 00H 7BH(Hex)
Para27		
Para28	Max Tag	Default:32 Reference:(decimal format) 10~64

7.2. TCPIP Parameter Table

Parameter	Description	Reference Value
Para1	IP Address (4 bytes)	Default:192.168.5.105 Example: IP = 192.168.5.105 Hex: C0 A8 05 69H
Para2		
Para3		
Para4		
Para5	Subnet mask (4 bytes)	Default:255.255.255.0 Example: SubNet Mask = 255.255.255.0 Hex: FF FF FF 00H
Para6		
Para7		
Para8		
Para9	Default gateway (4 bytes)	Default:192.168.5.1

Para10		Example: Gateway = 192.168.5.1 Hex: C0 A8 05 01H
Para11		
Para12		
Para13	IP Port (2 bytes)	Default:49152 Example: IP Port = 49152 Hex: C0 00H
Para14		
Para15	MAC Address (6 bytes)	Default:5E-45-A2-6C-30-1E Example: MAC = 5E-45-A2-6C-30-1E Hex: 5E 45 A2 6C 30 1EH
Para16		
Para17		
Para18		
Para19		
Para20		

7.3. Output Mode Parameter Table

Parameter	Description	Reference Value

Para1	Output Type	Default:0
		Reference:(decimal format)
		0-Decimal(1747988)
		1-Hex(1AAC14)
		2-Wiegand(02644052)

Para2	Output Bytes	Default:8 Reference:(decimal format) 8- 8(01747988) 9- 9(001747988) 10- 10(0001747988)
Para3	Is include Enter	Default:0 Reference:(decimal format) 0- Not include 1- Include