

13.56M RFID Reader YW-60x Serial

Manual

(version 1.6)

ISO15693

● English

<p>Web: http://www.yoworfid.com EMAIL: coodor@126.com Phone/Fax: +086-10-59395668 Mobile: +086-13671114914 Contact: Mr Zhou</p>
--

Content

1	Introduction.....	4
2	YW-605 Serial	4
3	Feature.....	4
4	Specifications	5
5	Program Manual.....	5
	DLL API and functions	5
	1. Get the version of DLL	5
	2. DES.....	5
	3. 3DES.....	6
	4. 3DES with vector.....	6
	5. Initial Serial Port.....	7
	6. Free Serial port.....	7
	7. Initial HID Port.....	8
	8. Free HID Port.....	8
	9. Change the serial port baud	8
	10. Set Reader ID.....	8
	11. Get Reader ID.....	9
	12. Get Reader Fireware version	9
	13. Get Reader serial No.	9
	14. Beep control.....	10
	15. LED control.....	10
	16. Set Content of 8 LED display	11
	17. Set the Status of Antenna	11
	18. Set Work Mode of Tags	11
	Functions about Tags of ISO15693	12
	19. 15693 Tags Inventory	12
	20. 15693 Tags Stay_Quiet.....	12
	21. Select Tag of ISO15693	13
	22. Reset To Ready of ISO15693 Tag	13
	23. Read Block of ISO15693 Tag	14
	24. Write Block of ISO15693 Tag	15
	25. Lock Block of ISO15693 Tags	15
	26. Write AFI of ISO15693 Tags	16
	27. Lock AFI of ISO15693 Tags	16
	28. Write DSFIC of ISO15693 Tags	17
	29. Lock DSFIC of ISO15693 Tags	18
	30. Get System Information of ISO15693 Tags	18
	31. Get Security of Blocks about ISO15693 Tag	19
	32. Multi Inventtory of ISO15693 Tags	20
	Functions of SAM	20
	33. SAM Baud.....	20
	34. SAM Reset.....	21
	35. Excute COS Command of SAM	21

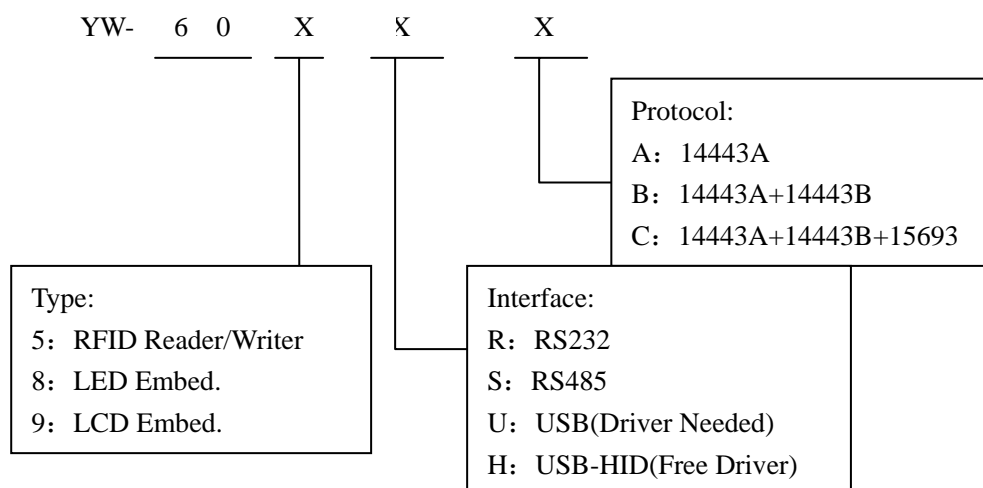
36.	SAM PPS Baud.....	22
6	Order Infomation.....	23

1 Introduction

YW-605 RFID reader/Writer is the 13.56M HF tags Reader and Writer. She Can support ISO14443A, ISO14443B and ISO15693 RFID tags.

2 YW-605 Serial

13.56M RFID Reader/Writer



3 Feature

- ☞ Distance of read tags is 5-10cm
- ☞ DLL API supported. VC, VB, Delphi, C++Builder, C#.net, WEB example
- ☞ Beep and LED embed, which can be controlled by API.

4 Specifications

- ☞ Baud: 19200BPS
- ☞ Power: DC5V \pm 10%
- ☞ Max Con: 1.5W
- ☞ Temp: -30°C \sim +70°C
- ☞ Humidity: 35% \sim 95%
- ☞ Size: 120 * 84 * 25 (mm)
- ☞ Weight: 100g

5 Program Manual

DLL API and functions

1. Get the version of DLL

prototype: `int stdcall YW_GetDLLVersion(void);`

Param: 无

Return: version Number is success, else is fail.

2. DES

prototype: `int stdcall DES(unsigned char cModel, unsigned char *pkey,
unsigned char *in, unsigned char *out);`

Param:

Param	Type	

cModel	unsigned char	Direction: 0 Encryption, 1 Decryption
pkey	unsigned char*	Key , 8 Bytes
in	unsigned char*	Data ,8 bytes
out	unsigned char*	Data after Des, 8bytes

Return: 0

3. 3DES

prototype: `int stdcall DES3(unsigned char cModel, unsigned char *pKey, unsigned char *In, unsigned char *Out);`

Param:

Param	Type	
cModel	unsigned char	Direction: 0 Encryption, 1 Decryption
pkey	unsigned char*	Key, 16Bytes
in	unsigned char*	Data ,8 bytes
out	unsigned char*	Data after Des, 8bytes

Return: 0

4. 3DES with vector

prototype: `int stdcall DES3_CBC(unsigned char cModel, unsigned char`

*pKey, unsigned char *In, unsigned char *Out, unsigned char *pIV);

Param:

Param	Type	
cModel	unsigned char	Direction: 0 Encryption, 1 Decryption
pkey	unsigned char*	Key, 16Bytes
in	unsigned char*	Data ,8 bytes
out	unsigned char*	Data after Des, 8bytes
pIV	unsigned char*	Vector , 8Bytes

Return: 0

5. Initial Serial Port

prototype: `int stdcall YW_ComInitial(int PortIndex, int Bound);`

Param:

Param	Type	
PortIndex	<code>int</code>	Port Number, 1--255
Bound	<code>int</code>	Baud, 2400—115200, 19200default

Return: 1 success, 0 fail

6. Free Serial port

prototype: `int stdcall YW_ComFree(void);`

Param:

Return: 1 success, 0 fail

7. Initial HID Port

prototype: `int stdcall YW_USBHIDInitial(void);`

Param:

Return: 1 success, 0 fail

8. Free HID Port

prototype: `int stdcall YW_USBHIDFree(void);`

Param:

Return: 1 success, 0 fail

9. Change the serial port baud

prototype: `int stdcall YW_ComNewBound(int ReaderID ,int NewBound);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID, default ID=0
NewBound	<code>int</code>	New Baud 0x01->9600bps 0x02->14400bps 0x03->19200bps 0x04->28800bps 0x05->38400bps 0x06->57600bps 0x07->115200bps

Return: 1 success, 0 fail

10. Set Reader ID

prototype: `int stdcall YW_SetReaderID(int OldID, int NewID);`

Param:

Param	Type	
OldID	<code>int</code>	Old Reader ID
NewID	<code>int</code>	New Read ID

Return: 1 success, 0 fail

11. Get Reader ID

prototype: `int stdcall YW_GetReaderID(int ReaderID);`

Param:

Param	Type	
ReaderID	<code>int</code>	ReaderID=0

Return: ≥ 0 success and is Reader ID, other is fail

12. Get Reader Fireware version

prototype: `int stdcall YW_GetReaderVersion(int ReaderID);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID

Return: ≥ 0 success and is version, other is fail

13. Get Reader serial No.

prototype: `int stdcall YW_GetReaderSerial(int ReaderID, char *ReaderSerial);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
ReaderSerial	<code>Char *</code>	Reader Serial, 8 Bytes

Return: 1 success, 0 fail

14. Beep control

prototype: `int stdcall YW_Buzzer(int ReaderID, int Time_ON, int Time_OFF, int Cycle);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Time_ON	<code>int</code>	Beep on time, unit:100ms
Time_OFF	<code>int</code>	Beep off time, unit:100ms
Cycle	<code>int</code>	The weeks of the beep on and off

Return: 1 success, 0 fail

15. LED control

prototype: `int stdcall YW_Led(int ReaderID, int LEDIndex, int Time_ON, int Time_OFF, int Cycle, int LedIndexOn);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
LEDIndex	<code>int</code>	LED index 01: Read 02: Green 04: Yellow
Time_ON	<code>int</code>	LED On Time, unit:100ms
Time_OFF	<code>int</code>	LED Off Time, unit:100ms
Cycle	<code>int</code>	The weeks of the Led on and off
LedIndexOn	<code>int</code>	The index of Led on last: 00: all off 01: Read 02: Green

		04: Yellow
--	--	------------

Return: 1 success, 0 fail

16. Set Content of 8 LED display

prototype: `int stdcall YW_LEDDisplay(int ReaderID, int Alignment, char *LEDText);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Alignment	<code>int</code>	Alignment 1: Left 2: Center 3: Right
LEDText	<code>Char *</code>	The string to be displayed The char can be below 0123456789AbCdEF.-

Return: 1 success, 0 fail

17. Set the Status of Antenna

prototype: `int stdcall YW_AntennaStatus(int ReaderID, bool Status);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Status	<code>bool</code>	True: Open RF Antenna False: Close RF Antenna

Return: 1 success, 0 fail

18. Set Work Mode of Tags

prototype: `int stdcall YW_SearchCardMode(int ReaderID, int SearchMode);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
SearchMode	<code>char</code>	Tags 0x41-----IS014443A 0x42----- IS014443B 0x31----- IS015693 0x53-----ST Serail 0x52-----AT88RF020

Return: 1 success, 0 fail

Functions about Tags of IS015693

19. 15693 Tags Inventory

prototype: `int stdcall YW_IS015693_Inventory(int ReaderID, unsigned char *PData, unsigned char *PLen);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
PData	<code>unsigned char *</code>	If Succes, Flag(1byte)+ UID(8Byte)
PLen	<code>unsigned char *</code>	Length of PData

Return: 1 success, 0 fail

20. 15693 Tags Stay_Quiet

prototype: `int stdcall YW_IS015693_Stay_Quiet(int ReaderID, unsigned char *PUIID);`

Param:

Param	Type	
ReaderID	int	Reader ID
PUID	unsigned char *	UID(8Byte) of Tags

Return: 1 success, 0 fail

21. Select Tag of IS015693

prototype: int stdcall YW_IS015693_Select(int ReaderID, unsigned char Model, unsigned char *PUID);

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	Model=0x00 Select Tag With Flag Model=0x02 Select Tag with UID
PUID	unsigned char *	UID(8Byte) of Tags

Return: 1 success, 0 fail

22. Reset To Ready of IS015693 Tag

prototype: int stdcall YW_IS015693_Reset_To_Ready(int ReaderID, unsigned char Model, unsigned char *PUID);

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	Model=0x00 Select Tag With Flag

		Model=0x02 Select Tag with UID
PUID	unsigned char *	UID(8Byte) of Tags

Return: 1 success, 0 fail

23. Read Block of ISO15693 Tag

prototype: int stdcall YW_ISO15693_Read(int ReaderID, unsigned char Model, unsigned char *PUID, unsigned char StartBlockID, unsigned char BlockNums, unsigned char *PData, unsigned char *PLen);

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	Model=0x00 Select Tag With Flag Model=0x02 Select Tag with UID
PUID	unsigned char *	UID(8Byte) of Tags
StartBlockID	unsigned char	Block ID Begin
BlockNums	unsigned char	Numbers of Blocks
PData	unsigned char *	Data read
PLen	unsigned char *	Lenth of PData

Return: 1 success, 0 fail

24. Write Block of ISO15693 Tag

prototype: `int stdcall YW_ISO15693_Write(int ReaderID, unsigned char Model, unsigned char *PUID, unsigned char BlockID, unsigned char DataLen, unsigned char *PData);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Model	<code>unsigned char</code>	This value depends on the card . Bit0=1 Select Tag With Flag Bit1=1 Select Tag with UID Bit2=1 depends on the Tags
PUID	<code>unsigned char *</code>	UID(8Byte) of Tags
BlockID	<code>unsigned char</code>	Block ID Begin
DataLen	<code>int</code>	Length of PData
PData	<code>unsigned char *</code>	Data will be written

Return: 1 success, 0 fail

25. Lock Block of ISO15693 Tags

prototype: `int stdcall YW_ISO15693_Lock_Block(int ReaderID, unsigned char Model, unsigned char *PUID, unsigned char BlockID);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Model	<code>unsigned char</code>	This value depends on the card .

		Bit0=1 Select Tag With Flag Bit1=1 Select Tag with UID Bit2=1 depends on the Tags
PUID	unsigned char *	UID(8Byte) of Tags
BlockID	unsigned char	The block ID

Return: 1 success, 0 fail

26. Write AFI of ISO15693 Tags

prototype: `int stdcall YW_ISO15693_Write_AFI(int ReaderID, unsigned char Model, unsigned char *PUID, unsigned char AFI);`

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	This value depends on the card. Bit0=1 Select Tag With Flag Bit1=1 Select Tag with UID Bit2=1 depends on the Tags
PUID	unsigned char *	UID(8Byte) of Tags
AFI	unsigned char	AFI Value

Return: 1 success, 0 fail

27. Lock AFI of ISO15693 Tags

prototype: `int stdcall YW_IS015693_Lock_AFI(int ReaderID, unsigned char Model, unsigned char *PUID);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Model	<code>unsigned char</code>	This value depends on the card. Bit0=1 Select Tag With Flag Bit1=1 Select Tag with UID Bit2=1 depends on the Tags
PUID	<code>unsigned char *</code>	UID(8Byte) of Tags

Return: 1 success, 0 fail

28. Write DSFIC of IS015693 Tags

prototype: `int stdcall YW_IS015693_Write_DSFI(int ReaderID, unsigned char Model, unsigned char *PUID, unsigned char DSFID);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
Model	<code>unsigned char</code>	This value depends on the card. Bit0=1 Select Tag With Flag Bit1=1 Select Tag with UID Bit2=1 depends on the Tags
PUID	<code>unsigned char *</code>	UID(8Byte) of Tags
DSFID	<code>unsigned</code>	DSFID Value

	char	
--	------	--

Return: 1 success, 0 fail

29. Lock DSFIC of ISO15693 Tags

prototype: `int stdcall YW_ISO15693_Lock_DSFIID(int ReaderID, unsigned char Model, unsigned char *PUIID);`

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	This value depends on the card. Bit0=1 Select Tag With Flag Bit1=1 Select Tag with UID Bit2=1 depends on the Tags
PUIID	unsigned char *	UID(8Byte) of Tags

Return: 1 success, 0 fail

30. Get System Information of ISO15693 Tags

prototype: `int stdcall YW_ISO15693_Get_System_Information(int ReaderID, unsigned char Model, unsigned char *PUIID, unsigned char *PData, unsigned char *PLen);`

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	Model=0x00 Select Tag With Flag Model=0x02 Select Tag with

		UID
PUID	unsigned char *	UID(8Byte) of Tags
PData	unsigned char *	System Infomation
PLen	unsigned char *	Length of System Infomation

Return: 1 success, 0 fail

31. Get Security of Blocks about ISO15693 Tag

prototype: `int stdcall YW_ISO15693_Get_Block_Security(int ReaderID, unsigned char Model, unsigned char *PUID, unsigned char StartBlockID, unsigned char BlockNums, unsigned char *PData, unsigned char *PLen);`

Param:

Param	Type	
ReaderID	int	Reader ID
Model	unsigned char	Model=0x00 Select Tag With Flag Model=0x02 Select Tag with UID
PUID	unsigned char *	UID(8Byte) of Tags
StartBlockID	unsigned char	Block ID Begin
BlockNums	unsigned char	Number of Blocks
PData	unsigned char *	Security Infomation
PLen	unsigned	Length of Security

	<code>char *</code>	Information
--	---------------------	-------------

Return: 1 success, 0 fail

32. Multi Inventtory of ISO15693 Tags

prototype: `int stdcall YW_ISO15693_Multi_Inventory(int ReaderID, unsigned char AFIEEnable, unsigned char AFI, unsigned char *PData, unsigned char *PLen);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
AFIEEnable	<code>unsigned char</code>	0
AFI	<code>unsigned char</code>	0
PData	<code>unsigned char *</code>	UIDs of many Tags
PLen	<code>unsigned char *</code>	Length of UIDs

Return: 1 success, 0 fail

Functions of SAM

33. SAM Baud

prototype: `int __stdcall YW_SAM_ResetBaud(int ReaderID, int SAMIndex, int BaudIndex);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
SAMIndex	<code>int</code>	SAM Index
BaudIndex	<code>int</code>	0x00->9600 (Default)

		0x01->19200
		0x02->38400
		0x03->55800
		0x04->57600
		0x05->115200

Return: 1 success, 0 fail

34. SAM Reset

prototype: `int __stdcall YW_SAM_Reset(int ReaderID, int SAMIndex, int *rtLen, unsigned char *pData);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
SAMIndex	<code>int</code>	SAM Index
rtLen	<code>int *</code>	Length of PData
pData	<code>unsigned char *</code>	Data return of Sam Reset

Return: 1 success, 0 fail

35. Excute COS Command of SAM

prototype: `int __stdcall YW_SAM_COS(int ReaderID, int SAMIndex, int LenCOS, unsigned char *Com_COS, int *rtLen, unsigned char *pData);`

Param:

Param	Type	
ReaderID	<code>int</code>	Reader ID
SAMIndex	<code>int</code>	SAM Index
LenCOS	<code>int</code>	Length of COS Command
Com_COS	<code>unsigned char *</code>	COS Command

rtLen	unsigned char *	Length of pData
pData	unsigned char *	Data return after execution of COS Command

Return: 1 success, 0 fail

36. SAM PPS Baud

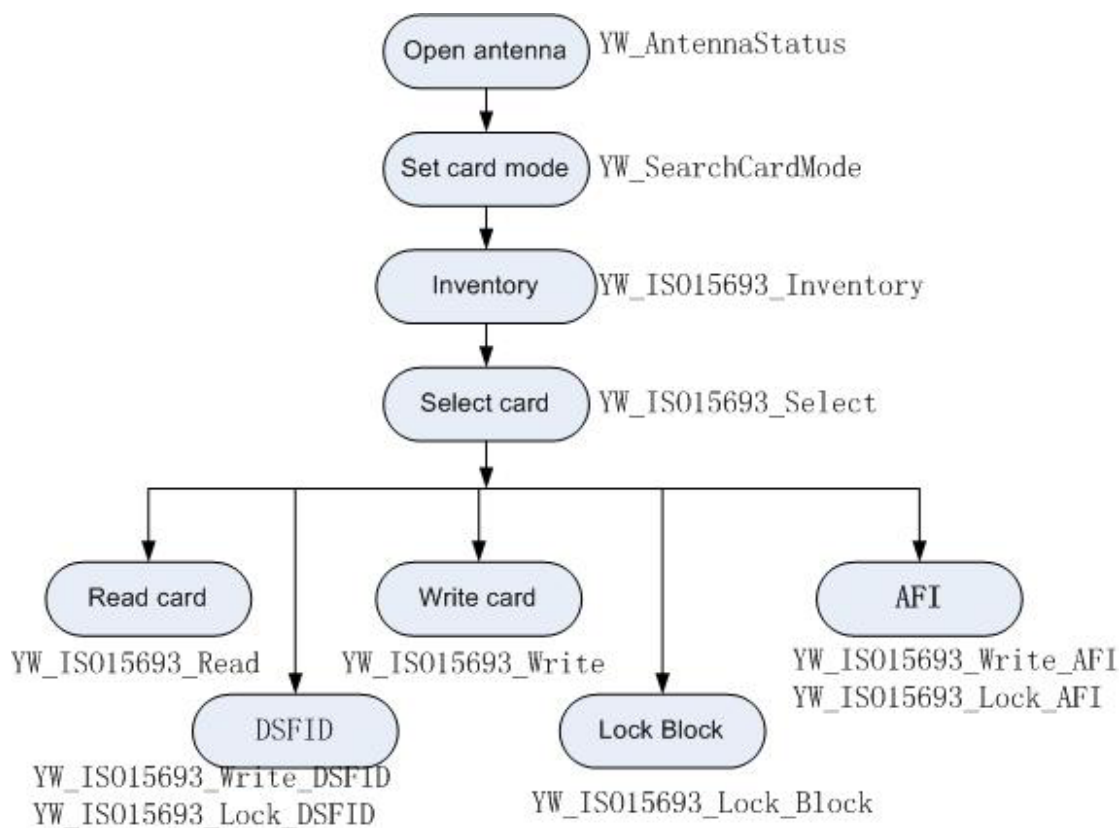
prototype: int __stdcall YW_SAM_PPSBaud(int ReaderID, int SAMIndex, int BaudIndex);

Param:

Param	Type	
ReaderID	int	Reader ID
SAMIndex	int	SAM Index
BaudIndex	int	0x00->9600 (Default) 0x01->19200 0x02->38400 0x03->55800 0x04->57600 0x05->115200

Return: 1 success, 0 fail

6 ISO15693 card operation



7 Order Infomation

YOWO RFID Technology Co.,LTD

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

Phone/Fax: 086-10-59395668

Mobile: 086-13671114914

Email: coodor@126.com

MSN: coodor@126.com