

# 13.56M RFID Reader QU-60x Serial

## Manual

(version 1.5)

ISO14443B

- English

**Web:** [www.quio-rfid.de](http://www.quio-rfid.de)  
**EMAIL:** [kontakt@quio-rfid.de](mailto:kontakt@quio-rfid.de)  
**Phone/Fax:** +49-(0)202-404329

## 目录

1	Introduction.....	4
2	QU-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 of ISO14443B.....	12
19.	Reset Type B Tags.....	12
20.	Excute COS Command of Type B Tags.....	12
	Function of AT88RF020 tags.....	13
21.	Key Check of AT88RF020.....	13
22.	Read block of AT88RF020 tags.....	13
23.	Write Data of AT88RF020 Tags.....	14
24.	Lock Block of AT88RF020 Tags.....	14
25.	Count of AT88RF020 tags.....	14
26.	Unselect Card of AT88RF020 tags.....	15
27.	Multi Read block of AT88RF020 tags.....	15
28.	Multi Write Block of AT88RF020 tags.....	15
	Functions of ST card, SR176, SR512, SR1X4K etc.....	16
29.	Active ST tags.....	16
30.	Deactive ST tags.....	16
31.	Get UID of SR176 Tags.....	17
32.	Read Block of SR176 Tags.....	17
33.	Write data of SR176 tags.....	17

---

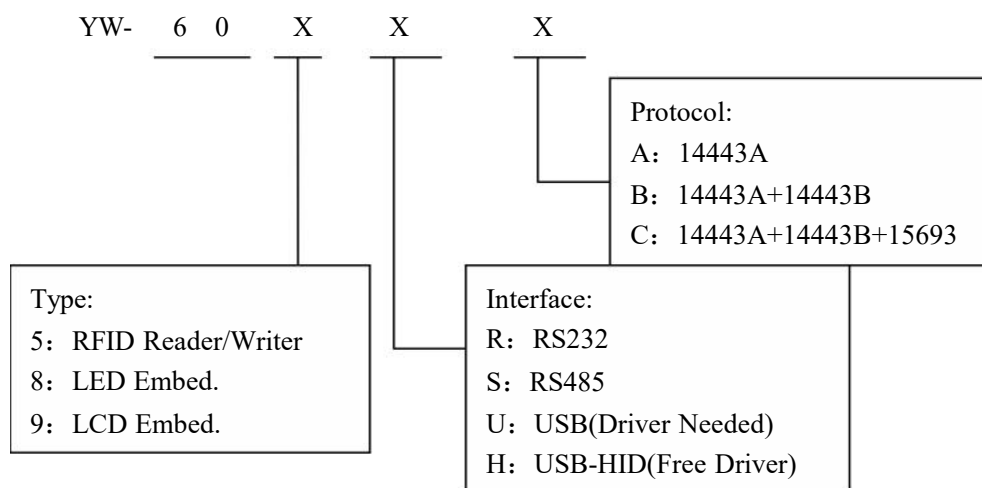
34.	Lock Block of SR176 Tags.....	18
35.	Get status of lock of SR176 tags.....	18
36.	Mutli read block of SR176 tags.....	18
37.	Multi Write block SR176 tags.....	19
38.	Get UID of SR512 Tags.....	19
39.	Read block of SR512 Tags.....	20
40.	Write Block of SR512 Tags.....	20
41.	Lock Block of SR512 Tags.....	20
42.	Get Lock Data of SR512 tags.....	21
43.	Multi Read Block of SR512 Tags.....	21
44.	Multi Write Block of SR512 tags.....	21
45.	Key Check of SRIX4K tags.....	22
	Functions of SAM.....	22
46.	SAM Baud.....	22
47.	SAM Reset.....	23
48.	Excute COS Command of SAM.....	23
49.	SAM PPS Baud.....	24
6	Order Infomation.....	25

# 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 RFIDReader/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	Char *	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	Char *	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 Antanna

**Return:**1 success, 0 fail

#### 18. Set Work Mode of Tags

```

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

```

**Param:**

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

**Return:** 1 success, 0 fail

Functions of IS014443B

#### 19. Reset Type B Tags

```

prototype: int stdcall YW_TypeB_Reset(int ReaderID, unsigned char
Mode, int *rtLen, unsigned char *pData);

```

**Param:**

Param	Type	
ReaderID	int	Reader ID
Mode	unsigned char	Request Mode 0x52----- All Card 0x26----- Active Card
rtLen	int *	Length of pData
pData	Char *	Reset information of reset

**Return:** 1 success, 0 fail

#### 20. Excute COS Command of Type B Tags

```

prototype: int stdcall YW_TypeB_COS(int ReaderID, int LenCOS,
unsigned char *Com_COS, int *DataLen, unsigned char *pData);

```

**Param:**

Param	Type	
ReaderID	int	Reader ID
Mode	unsigned char	Request Mode 0x52----- All Card 0x26----- Active Card
LenCOS	int	Length of COS Command
Com_COS	unsigned char*	COS Command
rtLen	int *	Length of pData
pData	Char *	Data returned after execute COS Command

**Return:** 1 success, 0 fail

Function of AT88RF020 tags

21. Key Check of AT88RF020

```

prototype: int stdcall YW_AT88RF020_Check(int ReaderID, unsigned char
*Key);

```

**Param:**

Param	Type	
ReaderID	int	Reader ID
Key	Char *	Key (8 bytes)

**Return:** 1 success, 0 fail

22. Read block of AT88RF020 tags

```

prototype: int stdcall YW_AT88RF020_Read(int ReaderID, int
BlockID, unsigned char *pData);

```

**Param:**

Param	Type	
ReaderID	int	Reader ID
BlockID	int	The block ID
pData	Char *	data read

**Return:** 1 success, 0 fail

## 23. Write Data of AT88RF020 Tags

**prototype:** int stdcall YW\_AT88RF020\_Write(int ReaderID, int BlockID, unsigned char \*pData);

**Param:**

Param	Type	
ReaderID	int	Reader ID
BlockID	int	block ID
pData	Char *	Data will be written

**Return:** 1 success, 0 fail

## 24. Lock Block of AT88RF020 Tags

**prototype:** int stdcall YW\_AT88RF020\_Lock(int ReaderID, unsigned char \*LockByte);

**Param:**

Param	Type	
ReaderID	int	Reader ID
LockByte	Char *	Lock data(4 bytes)

**Return:** 1 success, 0 fail

## 25. Count of AT88RF020 tags

**prototype:** int stdcall YW\_AT88RF020\_Count(int ReaderID, unsigned char \*Signature);

**Param:**

Param	Type	
ReaderID	int	Reader ID
Signature	Char *	Signature (6 bytes)

**Return:** 1 success, 0 fail

## 26. Unselect Card of AT88RF020 tags

**prototype:** int stdcallYW\_AT88RF020\_DeSel(int ReaderID);**Param:**

Param	Type	
ReaderID	int	Reader ID

**Return:** 1 success, 0 fail

## 27. Multi Read block of AT88RF020 tags

**prototype:** int stdcallYW\_AT88RF020\_ReadMulti(int ReaderID, int BlockID, int BlockNums, int \*LenData, unsigned char \*pData);**Param:**

Param	Type	
ReaderID	int	Reader ID
BlockID	int	Block ID
BlockNums	int	Number of Blocks
LenData	Int*	Length of pData
pData	unsigned char *	Data read

**Return:** 1 success, 0 fail

## 28. Multi Write Block of AT88RF020 tags

**prototype:** int stdcallYW\_AT88RF020\_WriteMulti(int ReaderID, int

BlockID, int BlockNums, int LenData, unsigned char \*pData);

Param:

Param	Type	
ReaderID	int	Reader ID
BlockID	int	Block ID
BlockNums	int	Number of Blocks
LenData	Int	Length of pData
pData	unsigned char *	Data will be written

Return: 1 success, 0 fail

Functions of ST card, SR176, SR512, SR1X4K etc.

### 29. Active ST tags

prototype: int stdcall YW\_ST\_Active(int ReaderID, unsigned char \*ChipID);

Param:

Param	Type	
ReaderID	int	Reader ID
ChipID	unsigned char *	Chip id if card (1 byte)

Return: 1 success, 0 fail

### 30. Deactive ST tags

prototype: int stdcall YW\_ST\_DeActive(int ReaderID);

Param:

Param	Type	
ReaderID	int	Reader ID

Return: 1 success, 0 fail



## 31. Get UID of SR176 Tags

**prototype:** `int` stdcall YW\_SR176\_GetUID(int ReaderID, unsigned char \*UID);

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
UID	unsigned char *	UID of SR176 (8 Bytes)

**Return:** 1 success, 0 fail

## 32. Read Block of SR176 Tags

**prototype:** `int` stdcall YW\_SR176\_Read(int ReaderID, int BlockID, unsigned char \*pData);

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
BlockID	<code>int</code>	Block ID
pData	unsigned char *	Data read (2 bytes)

**Return:** 1 success, 0 fail

## 33. Write data of SR176 tags

**prototype:** `int` stdcall YW\_SR176\_Write(int ReaderID, int BlockID, unsigned char \*pData);

**Param:**

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

pData	unsigned char *	Data will be written(2 bytes)
-------	--------------------	-------------------------------

**Return:** 1 success, 0 fail

#### 34. Lock Block of SR176 Tags

**prototype:** `int stdcall YW_SR176_Lock(int ReaderID, unsigned char LockByte);`

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
LockByte	unsigned char	Lock Data (1 byte)

**Return:** 1 success, 0 fail

#### 35. Get status of lock of SR176 tags

**prototype:** `int stdcall YW_SR176_LockStatus(int ReaderID, unsigned char *LockByte);`

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
LockByte	unsigned char*	Lock Data(1 byte)

**Return:** 1 success, 0 fail

#### 36. Mutli read block of SR176 tags

**prototype:** `int stdcall YW_SR176_ReadMulti(int ReaderID, int BlockID, int BlockNums, int *LenData, unsigned char *pData);`

**Param:**

Param	Type	
-------	------	--

ReaderID	<code>int</code>	Reader ID
BlockID	<code>int</code>	Block ID begin
BlockNums	<code>int</code>	Number of blocks
LenData	<code>Int*</code>	Length of pData
pData	unsigned char*	Data read

**Return:** 1 success, 0 fail

### 37. Multi Write block SR176 tags

**prototype:** `int` stdcall YW\_SR176\_WriteMulti(int ReaderID, int BlockID, int BlockNums, int LenData, unsigned char \*pData);

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
BlockID	<code>int</code>	Block ID begin
BlockNums	<code>int</code>	Number of blocks
LenData	<code>Int</code>	Length of pData
pData	unsigned char*	Data will be written

**Return:** 1 success, 0 fail

### 38. Get UID of SR512 Tags

**prototype:** `int` stdcall YW\_SR512\_GetUID(int ReaderID, unsigned char \*UID);

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
UID	unsigned char*	UID(8 byte)

**Return:** 1 success, 0 fail

### 39. Read block of SR512 Tags

**prototype:** `int stdcall YW_SR512_Read(int ReaderID, int BlockID, unsigned char *pData);`

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
BlockID	<code>int</code>	Block ID
pData	unsigned char*	Data read(4 bytes)

**Return:** 1 success, 0 fail

### 40. Write Block of SR512 Tags

**prototype:** `int stdcall YW_SR512_Write(int ReaderID, int BlockID, unsigned char *pData);`

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
BlockID	<code>int</code>	Block ID
pData	unsigned char*	Data will be written(4 bytes)

**Return:** 1 success, 0 fail

### 41. Lock Block of SR512 Tags

**prototype:** `int stdcall YW_SR512_Lock(int ReaderID, unsigned short LockByte);`

**Param:**

Param	Type	
-------	------	--

ReaderID	int	Reader ID
LockByte	unsigned short	Lock Data

**Return:** 1 success, 0 fail

#### 42. Get Lock Data of SR512 tags

**prototype:** int stdcallYW\_SR512\_LockStatus(int ReaderID, unsigned short \*LockByte)

**Param:**

Param	Type	
ReaderID	int	Reader ID
LockByte	unsigned short*	Lock Data

**Return:** 1 success, 0 fail

#### 43. Multi Read Block of SR512 Tags

**prototype:** int stdcallYW\_SR512\_ReadMulti(int ReaderID, int BlockID, int BlockNums, int \*LenData, unsigned char \*pData)

**Param:**

Param	Type	
ReaderID	int	Reader ID
BlockID	int	Block ID begin
BlockNums	int	Number of blocks
LenData	Int*	Length of pData
pData	unsigned char *	Data read

**Return:** 1 success, 0 fail

#### 44. Multi Write Block of SR512 tags

**prototype:** `int stdcallYW_SR512_WriteMulti(int ReaderID, int BlockID, int BlockNums, int LenData, unsigned char *pData)`

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
BlockID	<code>int</code>	Block ID begin
BlockNums	<code>int</code>	Number of blocks
LenData	<code>Int</code>	Length of pData
pData	unsigned char *	Data will be written

**Return:** 1 success, 0 fail

#### 45. Key Check of SRIX4K tags

**prototype:** `int stdcallYW_SRIX4K_Check(int ReaderID, unsigned char *Key, unsigned char *Signature)`

**Param:**

Param	Type	
ReaderID	<code>int</code>	Reader ID
Key	unsigned char*	Key (6 bytes)
Signature	unsigned char*	Out signature (3 bytes)

**Return:** 1 success, 0 fail

#### Functions of SAM

##### 46. SAM Baud

**prototype:** `int stdcall YW_SAM_ResetBaud(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

## 47. SAM Reset

**prototype:** int \_\_stdcall YW\_SAM\_Reset(int ReaderID, int SAMIndex, int \*rtLen, unsigned char \*pData);

**Param:**

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

**Return:** 1 success, 0 fail

## 48. 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	<code>unsigned char *</code>	Length of pData
pData	<code>unsigned char *</code>	Data return after execution of COS Command

**Return:** 1 success, 0 fail

#### 49. SAM PPS Baud

**prototype:** `int __stdcall YW_SAM_PPSBaud(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



## 6 Order Information

Quick-Ohm Küpper & Co. GmbH

WEB: [www.quio-rfid.de](http://www.quio-rfid.de)

Phone/Fax: +49-(0) 202-404329

Email: [kontakt@quio-rfid.de](mailto:kontakt@quio-rfid.de)