



RT200 USER GUIDE

(VERSION 3)

February 15

catalog

February 15catalog.....	0
1.Product introduction.....	4
1.1 Brief introduction.....	4
1.2 Features:.....	4
1.3 Range.....	5
1.4 Characters.....	6
1.5 Engine appearance.....	7
1.5.1 Front view.....	7
1.5.2 Top view.....	8
1.5.3 Bottom view.....	8
1.6 Interface.....	9
1.6.1 Connector.....	9
1.6.2 Ribbon Cable.....	9
1.6.3 Pin definition.....	10
1.7 Design notice.....	10
1.7.1 Components construction.....	10
1.7.2 Temperature.....	10
1.7.3 Window range.....	11
2、Configuration protocol.....	11
2.1 Brief introduction.....	11
2.2 Using configuration command.....	12
2.2.1 Configuration (For serial port).....	12
2.2.2 Communication protocol.....	12
2.2.3 Configuration and response protocol.....	13
2.2.4 Handshake.....	13
2.2.5 Get information.....	14
2.2.6 Start or stop decoding.....	19
2.2.7 Set operation mode.....	19

2.2.8 Set configuration mode.....	20
2.2.9 Set message-control mode.....	22
2.2.10 Settings operation.....	23
3. CRC source code.....	24
4. Reference For Circuit Design.....	25
Technical support	26

1.Product introduction

1.1 Brief introduction

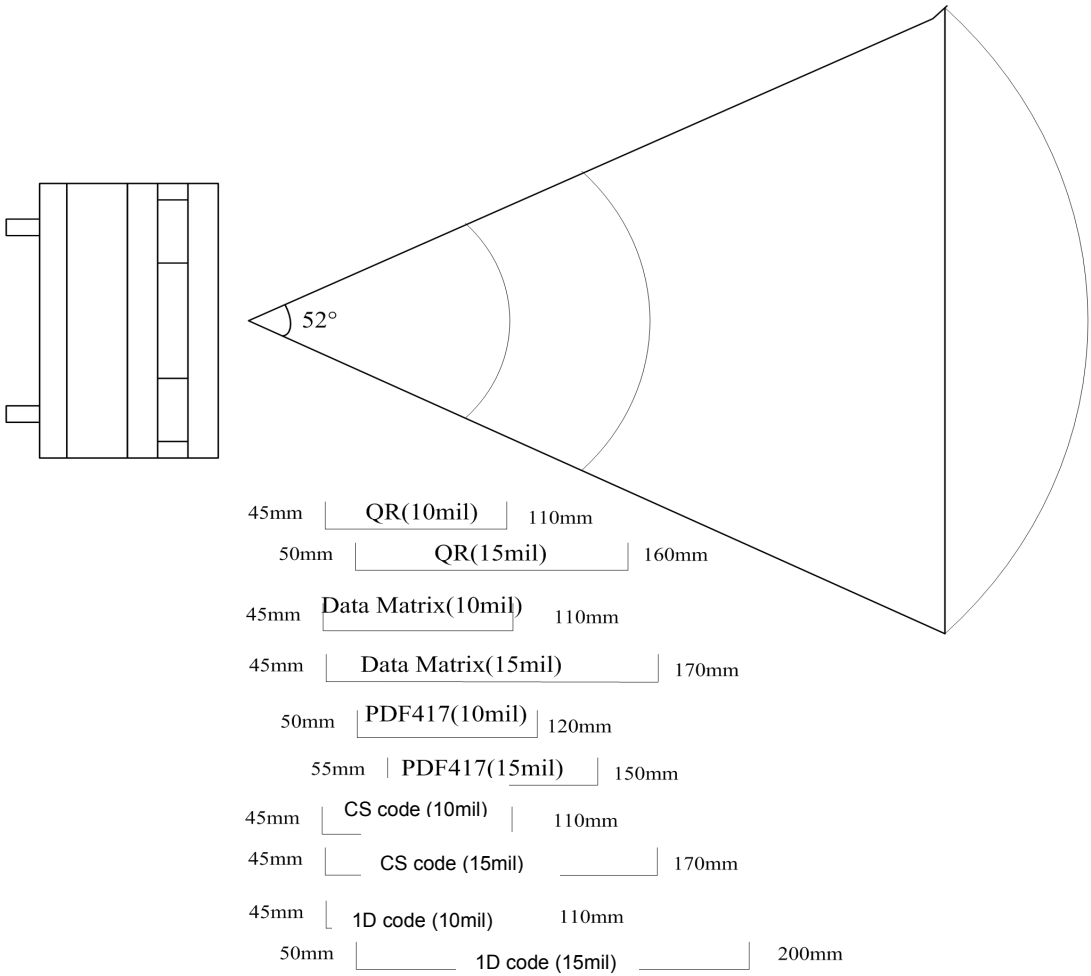
RT200 is an OEM image engine which Use CMOS image technology and advanced world level decoding algorithm, its a high performance engine that can decode most types of 1D/2D Barcode, and readable bar code either from paper or from LCD screen: phone screen, computer screen,etc.

With small and compact structure, RT200 can be easily integrated into all kinds of OEM terminals, including kiosks, ATM, PDA terminals and others portable devices. RTscan provides complete and detailed user guide and SDK documents for your integration work.

1.2 Features:

- Small structure, can be integrated into other devices easily
- Support most types of 1D and 2DBarcode
- With high performance processor, which makes decoding more fast and sensitive
- Can be configured easily, supporting firmware upgrading
- TTL interface (if use simulation kit together, then USB or RS232 interface)

1.3 Range



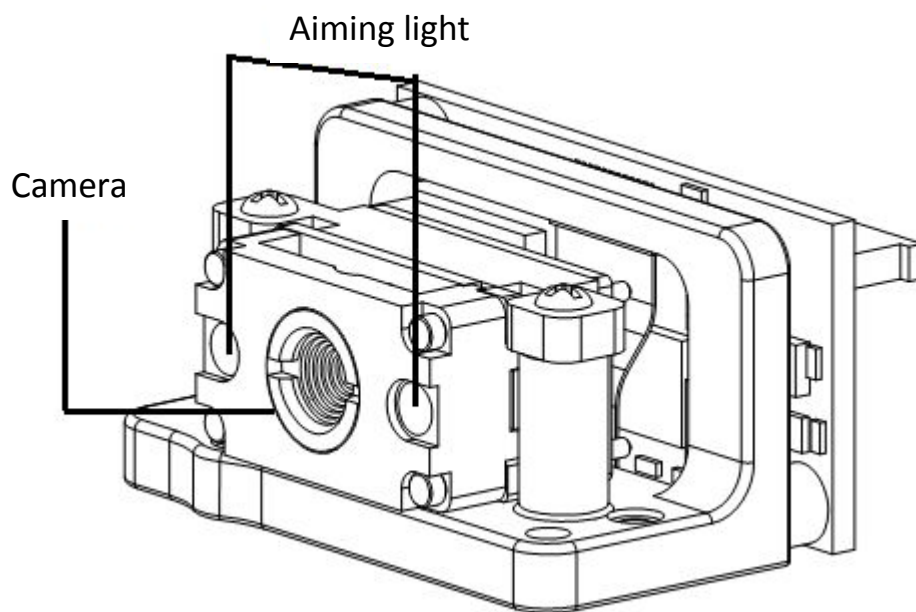
1.4 Characters

Decoding character		
Mode		CMOS
Sensor Resolution:		752x480
Symbologies	1D	PDF417, QR Code, Data Matrix, etc.
	2D	Code 128, EAN-13, EAN-8, Code39
Barcode accuracy		≥ 5mil
Focal Distance from Front of Engine		40mm~300mm
Min. Print Contrast		≥ 30%
Rotational Sensitivity		360° @ 0°Pitch and 0°Skew
Pitch Tolerance		60° @ 0°Roll and 0°Pitch
Skew Tolerance		60° @ 0°Roll and 0°Skew
Ambient Light		0 ~ 100,000 LUX
Electrical character		
Max. power consumption		1.4W
Voltage		3.3V
Current	Max	400 mA
	Operation	300 mA
	Standby	70 mA
Weight		25 g
Operation environment		
Operation temperature		-5℃ ~ +45℃
Storage temperature		-40℃ ~ +60℃
Operation humidity		5% - 95%

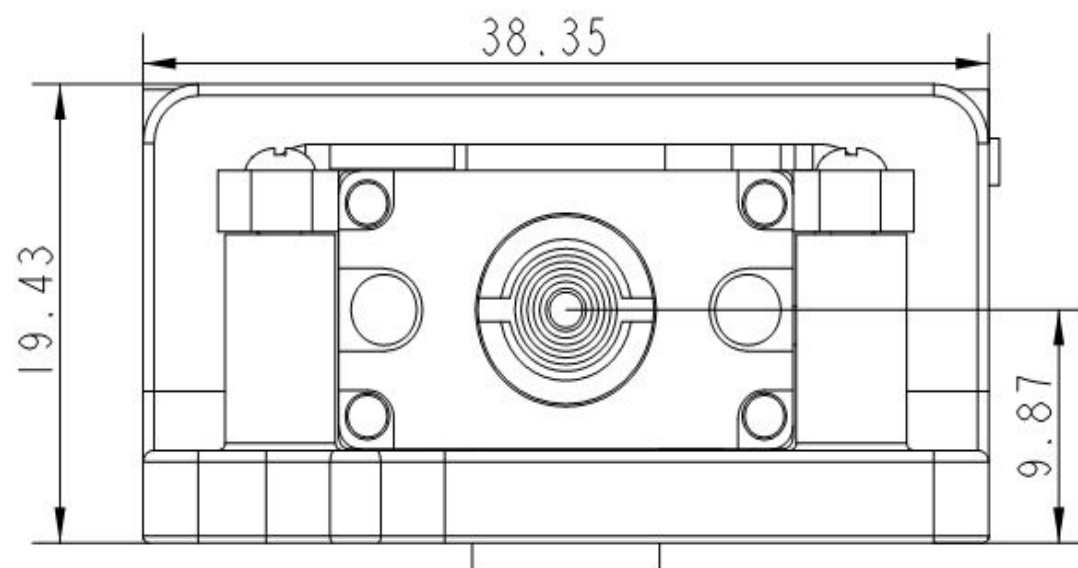
*Testing conditions

Code39, 4 Bytes ; min width = 10mil; width ratio = 3:1 ; PCS = 1.0 ; High = 12mm; distance= 100mm; temperature=23℃ ; ambient light= 200 LUX

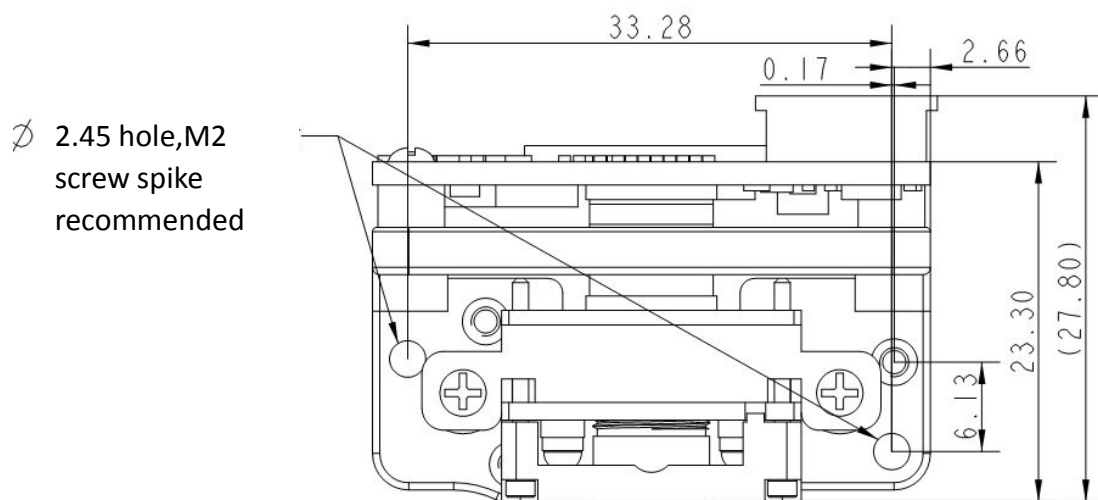
1.5 Engine appearance



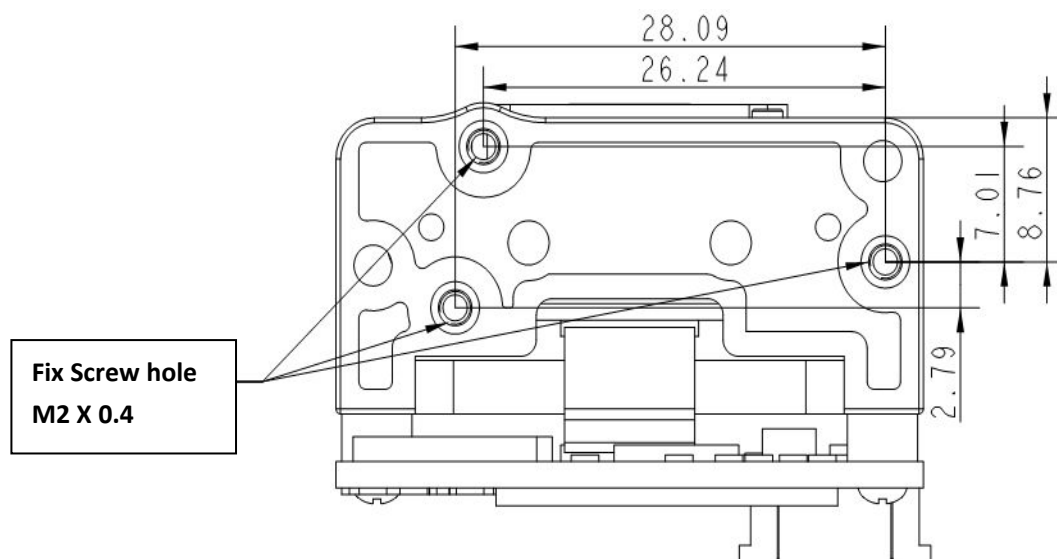
1.5.1 Front view



1.5.2 Top view



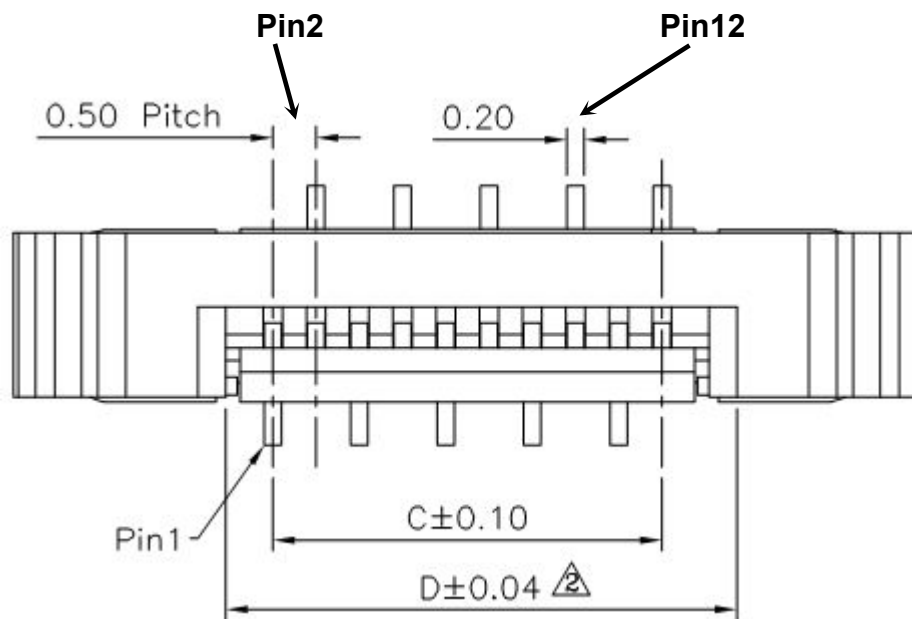
1.5.3 Bottom view



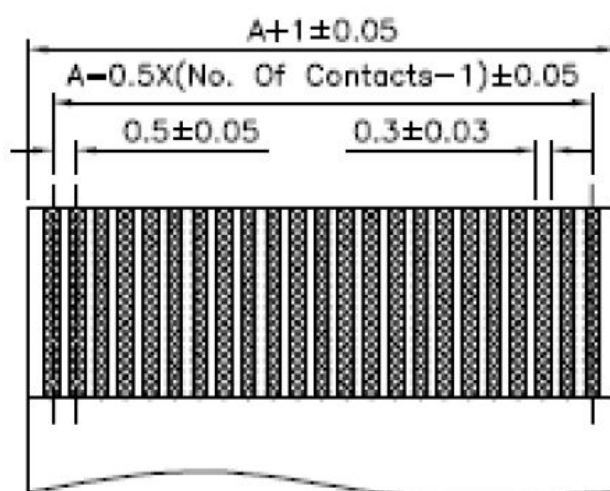
1.6 Interface

Connector and PIN out definition, as follows:

1.6.1 Connector



1.6.2 Ribbon Cable



FFC/FPC Dimension
Thickness 0.30 ± 0.03 mm 

1.6.3 Pin definition

PIN	input/output	Definition	description
PIN 1	—	—	—
PIN 2	source	VCC	+3.3V-5.5V
PIN 3	ground	GND	ground
PIN 4	input	RX	Serial input (reception from host system). TTL input levels. (Active High by a 10k resistor)
PIN 5	output	TX	Serial output to the host system . Sends TTL output levels.
PIN 6	—	NC	—
PIN 7	—	NC	—
PIN 8	—	NC	—
PIN 9	output	BUZ	Beeper control.
PIN10	output	LED	Indication led.
PIN11	input	POWERDOWN	Power on / off, power on the scanner when high level , and power off it when low level. Defaultly pull-up, and idle it if no use.
PIN12	input	TRIG	Trigger signal, Keep low at least 20 ms

You may also refer to the ***Interface and Integration of RT200.pdf*** to get some tips for integration via the TTL interface or via USB /RS232 with the emulation kit.

1.7 Design notice

1.7.1 Components construction

Leave enough space for RT200 and the cable, which make RT200 and the cable cannot be pressed or destroyed by any other components.

1.7.2 Temperature

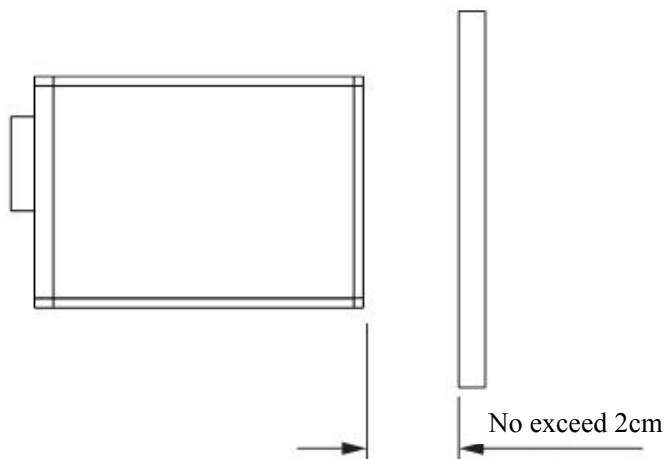
Under the continuous working, RT200's temperature will get higher, additional

cooling measure is recommended.

1.7.3 Window range

Some rules should be followed when using the engine:

- ✓ The engine's camera and LED should not be blocked.
- ✓ High transparent material should be used for better picture caption.
- ✓ window and the camera should be parallel, the distance between the window and the engine should be less than 2mm.



2、 Configuration protocol

2.1 Brief introduction

Two method can be used to configure the engine.

- ✓ Using configuration command, below 2.2 with introduction, and you can also refer to the [RT200 API Instruction.pdf](#), [RT200-Protocol-Instruction.pdf](#) and [Protocol Map.xls](#) for more information.
- ✓ Using setting codes which provided by : [Configuration Tool.exe](#) , to learn how to use the Configuration Tool.exe, please refer to the: [RT200-Configuration-tool-user-guide.pdf](#)

2.2 Using configuration command

2.2.1 Configuration (For serial port)

The engine uses the serial port as the communication interface, the default parameters is: 115200 baud rate, 8 data bits, no parity and stop bits.

2.2.2 Communication protocol

Host send command, and the engine response. According to the spec, developers can set or read the parameters of the engine. Note: the command and response follows the little Endian format.

2.2.2.1 Connect and Disconnect

Only for testing, to check whether engine can work properly.

2.2.2.2 Connect

This command tests the communication between host and engine.

Type	data	length	remark
Command	0x52(R)	1	Test if engine ready
response	0x59(Y)	1	OK

2.2.2.3 Disconnect

Type	data	length	remark
Command	0x4443(DC)	2	disconnect
response	No data		

2.2.3 Configuration and response protocol

Host can set or query the engine's parameter. host send command, and the engine response. According to the different command, host and engine complete different operation.

2.2.3.1 Command and response format

Command format: HEAD + LENTH +DATA +CRC

HEAD	LENTH	DATA	CRC
3bytes	2 bytes	Variable length	2bytes

HEAD: three fixed ASCCI char 'C', 'M', 'D'

LENTH: the length of the command,2 bytes;

DATA: Command body, including instruction code and corresponding data. The instruction is a five-char string. Command can be made by many sequential instructions. Only sending one instruction code is suggested.

CRC: CRC data of command.

Response format: HEAD + LENTH +DATA +CRC

HEAD	LENTH	DATA	CRC
3bytes	2 bytes	Variable length	2bytes

HEAD: three fixed ASCCI char 'R', 'S', 'P'

LENTH: response length, 2 bytes;

DATA: response body, including instruction code and corresponding data.

CRC: CRC of the command.

2.2.4 Handshake

Before sending command, host should send handshake first, in order to test the engine whether works normally. When received the right response, the host can then

send instruction.

handshake	data	Length(bytes)	remark
command	0x59(Y)	1	
response	0xAABB0AABB	4	

2.2.5 Get information

1) description:

host send this instruction to engine to get all kinds of engine's information ,including software and hardware information, operation mode, 1D/2D status.

2) format:

command: head+length+instruction code +CRC

response: "RSP" + length + data(information)+ CRC

instruction code:

instruction code (ASCII)	description
85505	Get hardware information
88306	Get software information
88501	Get operation mode
88603	Get 1D symbology status
88705	Get 2D symbology status
88807	Get the image stored in engine's memory

2.2.5.1 Get hardware information

Hardware information including hardware version、type、serial number、product date can be acquired through this command.

CMD: "CMD" + 0x00 + 0x05 + "85505" + CRC

RSP: "RSP" + Length of hardware information (2bytes) (0x00 + 0x37) + hardware information (55 ↑ bytes) + CRC

The structure body of hardware information is a 55-byte data, as follows:

```
Type def struct Hardware Info {
    unsigned char hardware Ver[10]; //hardware version
    unsigned char hardware Type[10]; //hardware type
    unsigned char specs[10]; //hardware specification
    unsigned char serial Num[10]; //serial number
    unsigned char product Date[15]; //product date
} Hardware Info;
```

2.2.5.2 get software information

Each symbology has its own version, there is no connection between these version. The version format is:xx.xx.xx with 21 bytes,the first 13 byte is the symbology, zero will be filled if less than 13 bytes. The following 8 bytes demonstrate the symbology version. Between each version, there will be two-byte data "\r\n".

CMD:"CMD" + 0x00 + 0x05 + "88306" + CRC

RSP: "RSP" + Length of software information (2bytes) + software information (variable length) + CRC

example: "QR Code: 01.02.24\r\nPDF417: 01.01.36\r\nDataMatrix:01.03.18\r\nCS Code: 01.01.10" , means that QR version is 01.02.24,PDF417 version01.01.36、Data Matrix version is01.03.18、CS 码 version is 1.01.10。

2.2.5.3 Get operation mode

CMD: "CMD"+0x00+0x05+"88501"+CRC

RSP: "RSP"+0x00+0x0F+current operation mode (15byte, each mode has 5 bytes)
+CRC

Operation mode includes decoding mode, supplementing light mode and location light mode. Each mode has its own command, and the sequence is not fixed, the operation mode can be determined by the command.

The commands are as follows:

Instruction code (ASCII)	description
Trigger mode	
20301	Key pressed
20302	Auto
20303	continuous (key pressed trigger first)
20304	Single auto
supplementing light(Red Led)	
20101	flash
20102	Always on
20103	Always off
Location light(Greed led)	
20201	flash
20202	Always on
20203	Always off

example: "203012010120201", "20301"means the current decoding mode is

key-press triggered, "20101" means supplementing light mode is flash mode, "20201" means the location light mode is flash mode.

2.2.5.4 Get 1D switch status

CMD: "CMD" + 0x00 + 0x05 + "88603" + CRC

RSP: "RSP" + 0x00 + 0x14 + 1D switch status (20bytes) + CRC

1D barcode includes 4 symbology: Code128、Code39、EAN-8、EAN-13, Each status has 5 bytes, so there are 20 bytes. The status sequence is not fixed, the 1D switch mode can be determined by the command.

The status are as follows:

table 0- 3 instruction code

Instruction code (ASCII)	description
1D	
10111	Enable code128
10110	Disable code128
10411	Enable code39
10410	Disable code39
10211	Enable EAN8
10210	Disable EAN8
10311	Enable EAN13
10310	Disable EAN13

example: "10111102111031110411", "10111" means enable code128, "10211" means enable EAN8, "10311" means enable EAN13, "10411" means enable code39.

2.2.5.5 : Get 2D switch status

CMD: "CMD"+ 0x00 + 0x05 + "88705"+ CRC

RSP: "RSP"+ 0x00 + 0x14 + 1D switch status (20bytes) +CRC

1D barcode includes QR ,CS,PDF417,Data Matrix, Each status has 5 bytes, so there are 20 bytes. The status sequence is not fixed, the 2D switch mode can be determined by the command.

The status are as follows:

2D	
10611	Enable QR
10610	Disable QR
10811	Enable CS
10810	Disable CS
10511	Enable PDF417
10510	Disable PDF417
10711	Enable Data Matrix
10710	Disable Data Matrix

example: "10611107111081110511", "10611"means enable QR, "10711" means enable Data Matrix,"10811" means enable CS,10511"means enable PDF417.

2.2.5.6: Get image stored in engine

CMD: "CMD"+0x00+0x05+"88807"+CRC

RSP: image data (length is 752*480)

2.2.6 Start or stop decoding

- 1) **description:** host send this instruction to engine, which make the engine to start or stop decoding

2) **format:**

command: head+length+instruction code +CRC

response: 0x59(Y)

3) **instruction code:**

instruction code (ASCII)	description
89910	Stop decoding
89811	Begin decoding

4) **example:** stop decoding

Step1: host send command: 0x59

Step2: received response: 0xAABB0AABB

Step3: host send command: "CMD"+0x0005+"89910"+(CRC)

Step4: received response: 0x59

2.2.7 Set operation mode

- 1) **description:** host send instruction code to set the operation mode.

2) **format:**

command: head +length+ instruction code +CRC

response: 0x59(Y)

3) **instruction code:**

Instruction code (ASCII)	description
Trigger mode	
20301	Key pressed
20302	Auto
20303	continuous (key pressed trigger first)
20304	Single auto
Red Led	
20101	flash
20102	Always on
20103	Always off
Green led	
20201	flash
20202	Always on
20203	Always off

4) example: set green led off

Step1: host send command: 0x59

Step2: received response: 0xAABB0AABB

Step3: host send command: "CMD"+0x0005+"20103"+(CRC)

Step4: received response: 0x59

2.2.8 Set configuration mode

1) **description:** host send instruction code to set the configuration mode.

2) **format:**

command: head +length+ instruction code +CRC

response: 0x59(Y)

3) instruction code:

Instruction code (ASCII)	description
1D	
10111	Enable code128
10110	Disable code128
10411	Enable code39
10410	Disable code39
10211	Enable EAN8
10210	Disable EAN8
10311	Enable EAN13
10310	Disable EAN13
2D	
10611	Enable QR
10610	Disable QR
10620	QR single code
10621	QR double code
10622	QR single or double code
10811	Enable CS
10810	Disable CS
10820	CS single code
10821	CS double code

10822	CS single or double code
10511	Enable PDF417
10510	Disable PDF417
10520	PDF417 single code
10521	PDF417 double code
10522	PDF417 single or double code
10711	Enable Data Matrix
10710	Disable Data Matrix
10720	Data Matrix single code
10721	Data Matrix double code
10722	Data Matrix single or double code

4) **example:** Disable QR

Step1: host send command: 0x59

Step2: received response: 0xAABB0AABB

Step3: host send command: "CMD"+0x0005+" 10610"+(CRC)

Step4: received response: 0x59

2.2.9 Set message-control mode

1) **description:** host send instruction code to set message-control mode.

2) **format:**

command: head +length+ instruction code +CRC

response: 0x59(Y) or head + length +message-control code +CRC

3) **instruction code:**

instruction code (ASCII)	description
-----------------------------	-------------

86101	enter message control mode
86110	exit message control mode
86201	Begin decoding
86202	response, format: result flag(1byte)+decoding data result flag: 1success 2timeout 3fail

4) **example:** exit message control mode

Step1: host send command: 0x59

Step2: received response: 0xAABB0AABB

Step3: host send command: "CMD"+0x0005+" 20302"+(CRC)

Step4: received response: 0x59

2.2.10 Settings operation

1) **description:** host send instruction code to save or restore engine's settings.

2) **format:**

command: head +length(0x05,2bytes)+ instruction code +CRC

response: 0x59(Y)

3) **instruction code:**

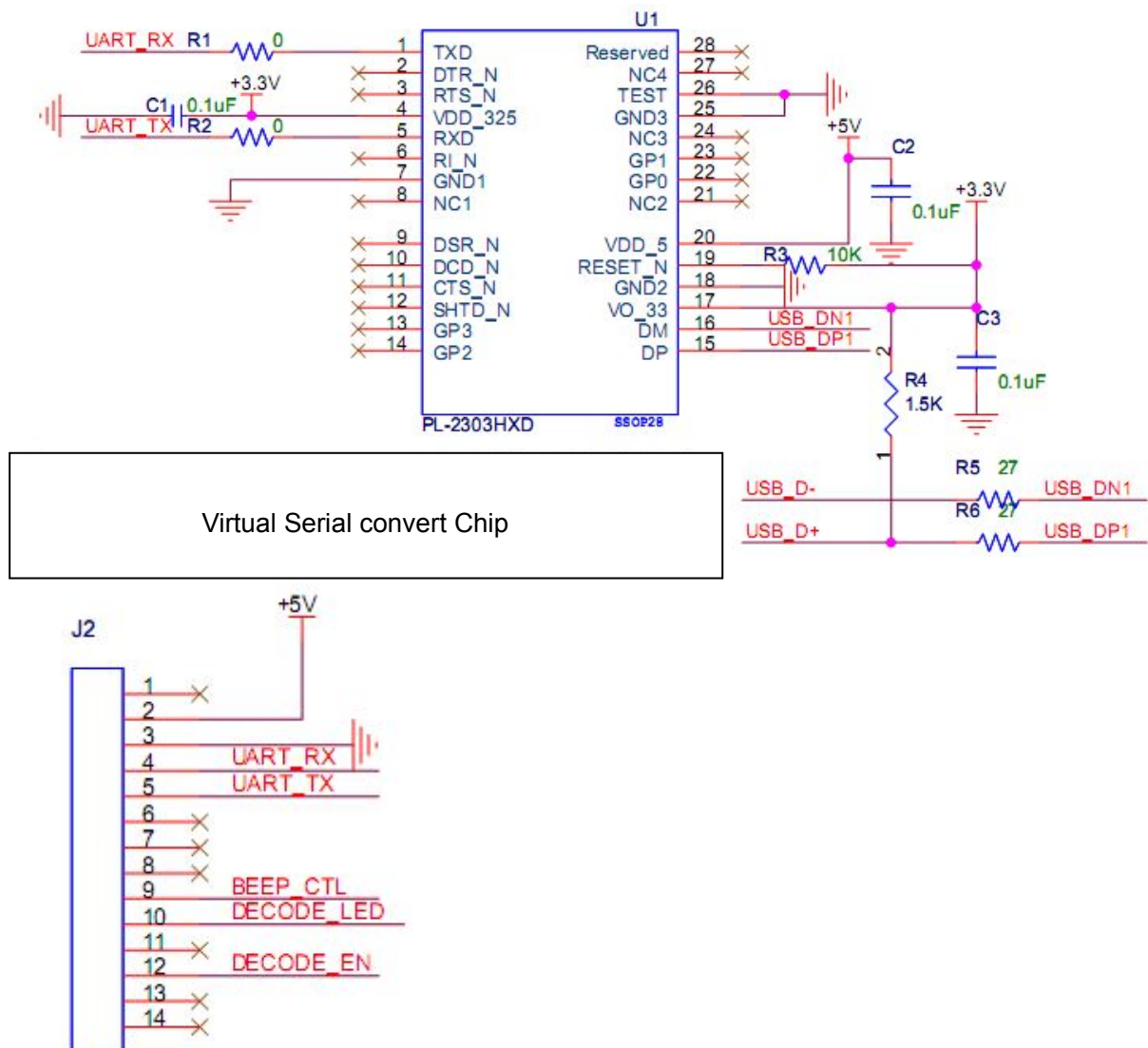
instruction code (ASCII)	description
89101	save user setting
89203	load user default settings
89305	restore default settings

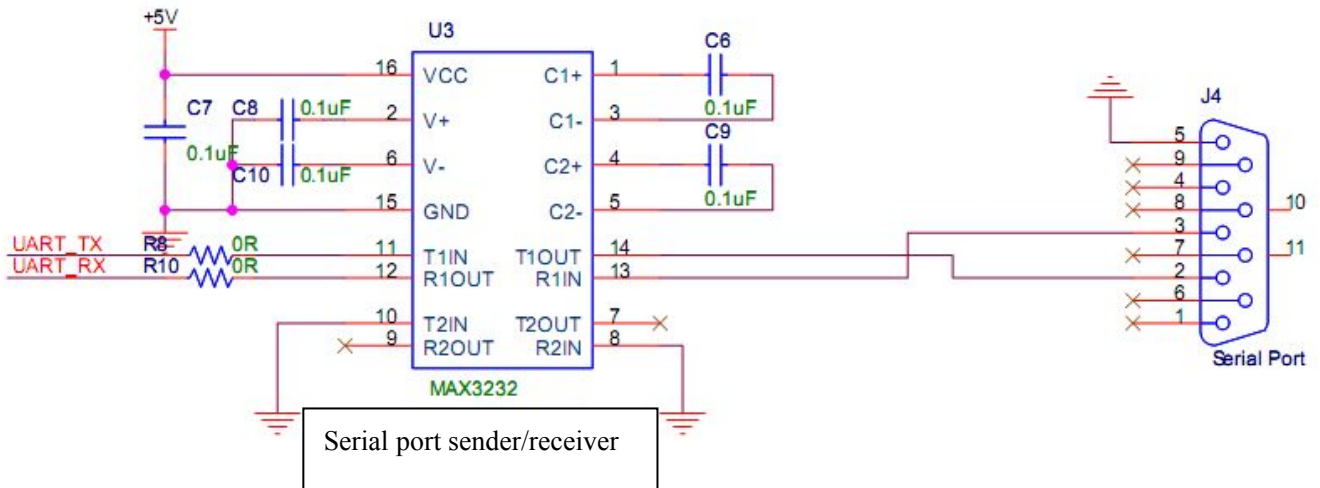
3. CRC source code

```
unsigned short cal_crc2(unsigned char *ptr, unsigned long len)
{
    unsigned char i;
    unsigned int crc=0;
    while(len--!=0)
    {
        for(i=0x80; i!=0; i/=2)
        {
            if((crc&0x8000)!=0)
            {
                crc*=2;
                crc^=0x18005;
            }
            else
            {
                crc*=2;
            }
            if((*ptr&i)!=0) crc^=0x18005;
        }
        ptr++;
    }
    return crc; }
```


4. Reference For Circuit Design

Refer to the following picture for hardware circuit design. The circuit includes the decoding button, a buzzer and a decoding LED, RS232 interface circuit and USB virtual Com circuit. For the USB virtual Com circuit, we use Prolific's PL2303.





For more information, please refer to the : [Demo Circuit diagram for hardware design.pdf](#), You may also refer to the [Interface and Integration of RT200.pdf](#) to get some tips for integration via the TTL interface or via USB /RS232 with the emulation kit.

Technical support

If need any technical support, feel free to contact our technical support department:

Support Department



RTSCAN TECHNOLOGY LIMITED

Add: F/7, Block C, Ying Da Li Park, Futian Tax-Free Trade Zone, Shenzhen, China

Tel: +86-755-32909260

E-mail: support@rtscan.net