

MOD1058

## IMPRIMANTE THERMIQUE AVEC INTERFACE SERIE TTL DFR0503-EN



### Introduction

---

The embedded thermal printer uses thermal technology, it doesn't need to add toner or ink, stable performance and mini size. The embedded thermal printers are widely used in instrumentation, food inspection, shopping terminals and other scenarios. It can be found in the ATM machines, lockers, POS and supermarket electronic scales in the life. The printer uses a 5~9V wide voltage supply mode. The higher the voltage, the clearer the printing effect. It is suitable for thermal paper rolls with a width of 58MM and able to provide a print life of 50 kilometers. It supports Arduino, Raspberry Pi, and Windows System. We can refer to the instruction set and use the USB serial port or TTL communication interface to send corresponding commands to the thermal printer. Then the printer will instantly print out the text, characters, barcode, QR code and even black and white pictures according to the instructions; we can connect it to the Arduino controller or the Raspberry Pi by a serial port. Combining with various sensors, we can DIY all kinds of practical instruments such as electronic scale, blood pressure index measuring instrument, soil element analyzer and driving recorder. Besides, we can also make a small ticket printer of our own. The content on the small ticket can be defined by ourselves. After further development, we can even realize the remote printing function. soil element analyzer and driving recorder. Besides, we can also make a small ticket printer of our own. The content on the small ticket can be defined by ourselves. After further development, we can even

realize the remote printing function.soil element analyzer and driving recorder. Besides, we can also make a small ticket printer of our own. The content on the small ticket can be defined by ourselves. After further development, we can even realize the remote printing function.



#### NOTE :

- The higher voltage of the embedded thermal printer, the faster printing speed, and the clearer handwriting of the printed documents (voltage range is 5~9V). At the same time, the normal operating current is between 0.8A and 1.5A. The instantaneous current can reach up to 2A. If the weather is cold, the operating current will need to be increased accordingly. The power consumption for printing graphics is generally large. If the printing ticket involves graphics, it is recommended to select the appropriate power supply.
- To print characters like Chinese with Arduino, you need to create a C file, and it cannot be compiled under Arduino IDE (Chinese character cannot be input under the IDE development environment. The printout will be garbled even if you use the clipboard to paste these characters into input end). Then write the relevant array data into C file for calling.
- USB interface supports computer system such as XP, win7, win8, and you need to install the driver on them before you can use it.

#### Specification

---

- Operating Voltage: 5 ~ 9V
- Operating Current: 0.8 ~ 1.5A (instantaneous current  $\approx 2A$ )
- Communication interface: TTL + USB
- Printing method: thermal dot matrix printing
- Print color: black and white output
- Resolution: 203 DPI (8 points / mm, 384 points per line)
- Use consumables: thermal ticket roll
- Paper roll specifications: paper roll width (58mm  $\pm$  0.5mm), paper roll diameter (30mm)
- Print content: support printing of text, graphics, 1D barcode, QR code, curve, etc.
- Print width: 48mm (effective print width)
- Print life: 50 km
- Printing speed: 60mm/s
- Print characters: GB18030 Chinese character library (24 24 *international first and second level font*), 12 24 standard ASCII code, and can print 1-4 times for Chinese characters and characters
- Working temperature: -20 ° C ~ 50 ° C
- Relative humidity: 10-80%

- Print order: ESC/POS compatible instruction set (see "Printer Instruction Set" for details)
- Appearance size: 82 x 58 x 44 mm/3.23x2.28x1.73"
- Installation size: 77 x 53 x 42 mm/3.03x2.08x1.65"

## Pinout

---



### TTL Communication Interface

Number	Name	Description
1	GND	Ground
2	DTR	Flow Control
3	RXD	Receive Data
4	TXD	Transmit Data
5	NC	Not Connected

### Power Interface

Number	Name	Description
1	VCC	5-9V

Number	Name	Description
2	GND	Ground

## Tutorial

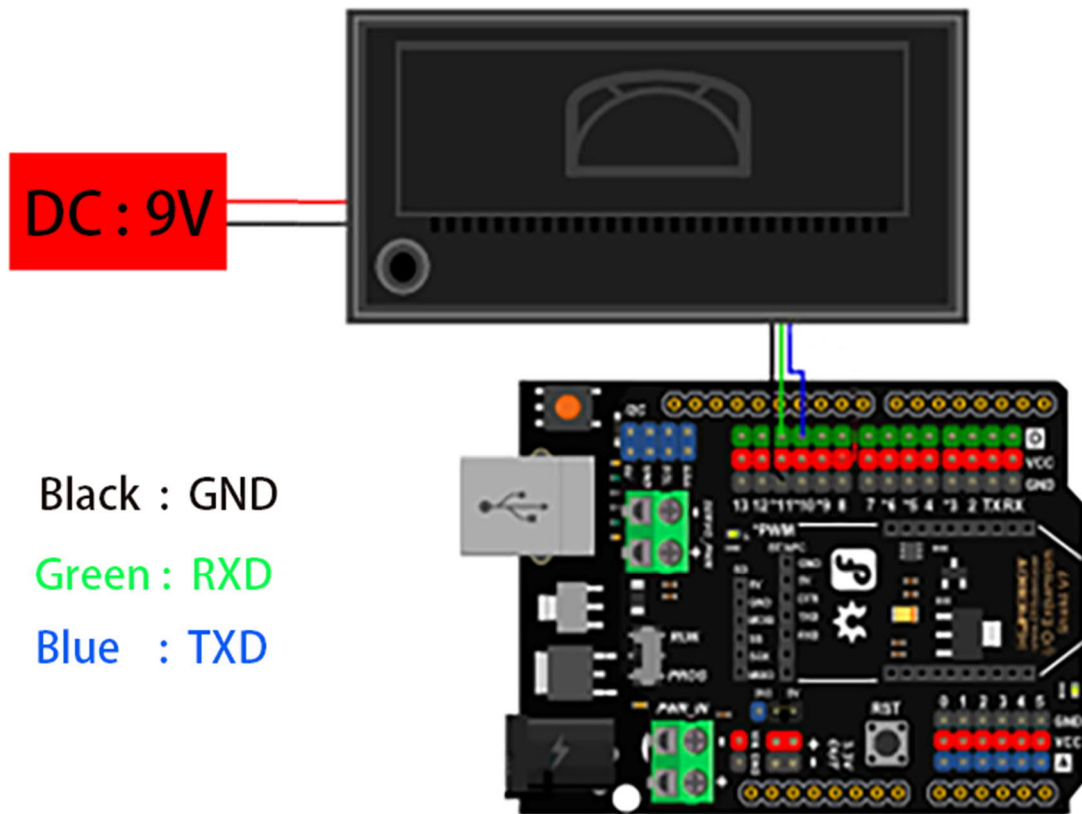
---

### Preparation

- Hardware
  - Arduino UNO Board ×1
  - IO Expansion Shield for Arduino V7.1 ×1
  - Power Supply DC ×1
  - Embedded Thermal Printer ×1
  - Connectors
- Software
  - Arduino IDE, Click to download [Arduino IDE](#)

### Connection Diagram

- DC power provides 9V power for Arduino UNO and Thermal Printer as shown below. Meanwhile, connect TTL interface of the printer to software serial port (10-RXD, 11-TXD) defined in the sample program of this wiki.



- Put the thermal paper into the printer.



## Print Character(Arduino)

- Use "print" statement to directly print non-Chinese character such as English letter, number, punctuation....
- Print Chinese character as the way shown below:

1.Create a hz.c file, the content should be a created character array. (How to create a .c file? Create a .txt file and revise the suffix to .c)

`const char hz[][32]` is a statement of a character string array in fact. One string cannot exceed 32 bytes (16-bit Chinese character coded in GB2312). If you want to display a long Chinese character, you need to revise the number "32"; actually, in other words, it means to take out all Chinese characters and put them into a array. An international program version can be easily realized by this way since we can display different language by just switching the array;

Note : The creation of this file aims to solve the problem that GB2312 Chinese character internal code cannot be edited under IDE Programming Environment. So don't open the file in IDE programming environment in case the file is rewritten into UTF8 internal code file.

```

const char hz[][32]={
"DFROBOT",           //0
"极趣科技",          //1
"嵌入式热敏打印机",  //2
"arduino开发控制",   //3
"DIY趣味",           //4
"让我们一起共创未来", //5
"打造我们的DIY世界",  //6
};

```

## 2.Compose Program

```

#include"C:\Users\Administrator\Documents\Arduino\uart_dayinji\hz.c"
//Reference the hz.c file created before, the content of the quotation should be
the index address of the hz.c file, and it need to be revised according to hz.c
actual address.
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
void setup()
{
    mySerial.begin(9600);
}
void loop()
{
    mySerial.println();
    mySerial.println();
    mySerial.println("DFROBOT");
    mySerial.println();
    mySerial.println(hz[0]);
    mySerial.println(hz[1]);
    mySerial.println(hz[2]);
    mySerial.println(hz[3]);
    mySerial.println(hz[4]);
    mySerial.println(hz[5]);
    mySerial.println(hz[6]);
    mySerial.println();
    mySerial.println();
    mySerial.println();
    delay(10000);
}

```

3.The following characters will be displayed on the thermal ticket roll.

```

DFROBOT
极趣科技
嵌入式热敏打印机
arduino开发控制
DIY趣味
让我们一起共创未来
打造我们的DIY世界`

```

## Print QR Code and Barcode(Arduino)

Write the print instructions of QR code and barcode in accordance with the reference of General Instruction Set and Special Instruction Set .

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX

/*-----Print barcode instruction, including information "DFR0503"-----*/
char bar_code[24]={
    0x1b, 0x40,
    0x1d, 0x48, 0x02, //the 43rd command of the general instruction, select the
    print address of the information (HRI character)included in the barcode.0x02 means
    that the printed information should be below the barcode.
    0x1d, 0x68, 0x64, //set barcode height, range"1~255", here we set it to "0x64"
    .
    0x1d, 0x77, 0x03, //the 45th command of the general instruction, set horizontal
    width, here we select "0x03".
    0x1d, 0x6b, 0x49, 0x09, 0x7B, 0x42, 0x44, 0x46, 0x52, 0x7B, 0x43, 0x05, 0x03
    //set the information included in barcode. Barcode length "0x09" correspond to
    "DFR0503". "0x44, 0x46, 0x52" are ACILL codes of the three letter D,F,R
    respectively. "0x05, 0x03" is "0503".
}; //if the information is "DFR050308", the barcode length should be"0x0B",
and so on.

/*-----Print QR code instruction, including information"www.dfrobot.com"-----
--*/
const char QRx[60]={
    0x1b,0x40,
    0x1d,0x28,0x6b,0x03,0x00,0x31,0x43,0x05, //Refer to the 1st command of special
    instruction set.QR code size "0x05"
    0x1d,0x28,0x6b,0x03,0x00,0x31,0x45,0x30, //Set QR code verification level to L.
    0x1d,0x28,0x6b,0x12,0x00,0x31,0x50,0x30,0x77,0x77,0x77,0x2E,0x64,0x66,0x72,0x6F,0x
    62,0x6F,0x74,0x2E,0x63,0x6F,0x6D, // "0x12" is the number of the character(15) of
    the information"www.dfrobot.com" that was included in QR code plus 3.
    //"0x77,0x77,0x77,0x2E,0x64,0x66,0x72,0x6F,0x62,0x6F,0x74,0x2E,0x63,0x6F,0x6D" are
    the ACILL code of the characters of the website address.
    0x1b,0x61,0x01, //centre the graphic
    0x1d,0x28,0x6b,0x03,0x00,0x31,0x52,0x30, //Verify the QR code
    0x1d,0x28,0x6b,0x03,0x00,0x31,0x51,0x30 //print the QR code
};
void setup(){
    mySerial.begin(9600);
    Serial.begin(9600);
    while(Serial.read()>= 0){}
}
void loop(){
    mySerial.println();
    mySerial.print("-----");
    mySerial.println();
    mySerial.write(bar_code,60); //send "print barcode instruction" to printer
    delay(5);
    mySerial.println();
    mySerial.write(QRx,60); //send "print QR code instruction" to printer.
    mySerial.println();
    mySerial.print("-----");
    delay(5000);
```



## Print Bitmap

- Refer to the 35th command of the official general instruction set "define the download bitmap mode"

1. Use software PCtoLCD2002 to abstract character matrix from a bitmap. (the size of the bitmap should be:  $x < 255$ ,  $y < 488$ ; it would be better if the bitmap size is the multiple of 8; set negative code, one by one, straightforward);

2. Create an "image.h" file, paste the abstracted hexadecimal array into it, and then you will get a bitmap character matrix library.

[illegible]

```

0x07, 0xC3, 0xE0, 0x0F, 0x83, 0xE0, 0x0F, 0x83, 0xE0, 0x0F, 0x83, 0xE0, 0x0F, 0x83, 0xE0, 0x0F, 0x
81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F,
0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF8, 0x0F, 0x80, 0xF8, 0x0F, 0x80, 0xF8, 0x0F, 0x80, 0x
F8, 0x0F, 0x80, 0xF8, 0x0F, 0x81, 0xF8, 0x0F, 0x81, 0xF0, 0x0F, 0x81,
0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x
0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x
0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x
81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0x
F0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F,
0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0x
F0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81,
0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xF0, 0x0F, 0x81, 0xE0, 0x0F, 0x83, 0xE0, 0x0F, 0x83, 0xE0, 0x
0F, 0x83, 0xE0, 0x07, 0xC3, 0xE0, 0x07, 0xFF, 0xE0, 0x07, 0xFF, 0xE0,
0x07, 0xFF, 0xC0, 0x03, 0xFF, 0xC0, 0x01, 0xFF, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x
00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x
00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x
00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x
00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0d, 0x0a,
0x1B, 0x40,
0x1b, 0x2a, 0x00, 0x0C, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x
FF,
//Print a division line
0x1B, 0x33, 0x00,
0x0A,
0x1D, 0x2F, 0x30
//Print bitmap instruction
};

```

### 3.Compose program and send instruction

```
#include <SoftwareSerial.h>
#include "D:\image.h" //Reference the created image.h file, and revise address
according to the actual condition.
SoftwareSerial mySerial(10, 11); // RX, TX

/*-----Print barcode instruction, including "DFR0503"-----*/
char picture_S[6]={
    0x1b, 0x40,
    0x1d, 0x2A, 0x14, 0x0A //Refer to general instruction set 34 and 35, get 0x14
and 0x0A by calculating bitmap size.
};
char picture_E[3]={
    0x1D, 0x2F, 0x30
};

void setup(){
    mySerial.begin(9600);
    Serial.begin(9600);
    while(Serial.read() >= 0){}
}

void loop(){
    mySerial.println();
    mySerial.print("-----");
    mySerial.println();

    mySerial.write(show, 2237); //Send "print bitmap instruction" to printer

    mySerial.println();
```

```
mySerial.print("-----");  
delay(5000);  
}
```

## Sample Code

---

Input "Mike" to Serial Port Assistant, after sending it to uno, the printer will print "Name: Mike"; when sending "1520000 \*\* ", it will be recognized as a telephone number, and the printer will print "Tel: 1520000\*\* "; Input a words beginning as : " this is a test", it will be recognized as a remark, and the printer will print "Remarks:"

```
#include "SoftwareSerial.h"  
#include "Wire.h"  
char buffer[30];  
SoftwareSerial mySerial(10, 11); // RX, TX  
String comdata = "";  
char comdataX;  
int numdata;  
void setup()  
{  
    mySerial.begin(9600);  
    Serial.begin(9600);  
    while(Serial.read() >= 0){}  
}  
void loop()  
{  
    //Serial_pc();  
    Serial_pc1();  
}  
void Serial_pc()  
{  
    while(Serial.available() > 0){  
        delay(100);  
        mySerial.print("DFROBOT: ");  
        numdata = Serial.readBytes(buffer, 30);  
        mySerial.print(buffer);  
    }  
    for(int i=0; i<30; i++)  
        buffer[i]=0;  
}  
void Serial_pc1()  
{  
    while(Serial.available() > 0){  
        delay(100);  
        numdata = Serial.readBytes(buffer, 30);  
  
        if(buffer[0] > '/' && buffer[0] < ':' )  
        {  
            mySerial.print("Tel: ");  
            mySerial.println(buffer);  
        }  
        else if(buffer[0] > '@' && buffer[0] < '{' )  
        {  
            mySerial.print("Name: ");  
            mySerial.println(buffer);  
        }  
    }  
}
```

```
else if(buffer[0]=='*')
{
    mySerial.print("Remarks:");
    mySerial.println(buffer);
}
else{;}
}
for(int i=0; i<30; i++)
buffer[i]=0;
}
```