Required Materials:

Hardware circuit: (1) ranging module; (2) USB rotary TTL module; (3) 3V or 3.3V power supply (I am powered by the 3V power supply brought with me on the purchased Learning Board); (4) Bread line several Software: (1)

Software：

（1）



If the computer has this driver, it will not have to be installed, mainly used in interface circuits.

（2）



This software is used for ranging output. View output results on your computer

Circuit Connection:

The first need to weld the circuit is as follows:



Negative connects to USB GND.

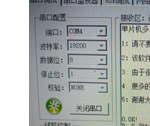
Just four wires, after the welding is complete. The 3V power connection is as the picture.

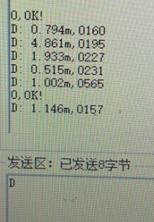
follows

USB conversion to TTL circuit connection is as follows:

This should pay attention to TXD Rx,rxd pick up Tx,usb transfer TTL module should also be grounded. In total, it adds up to five wires.

Install Porthelper on your computer and open it

Set parameters, pay attention to a little COM4 choice due to different computer interface and different, in general, one can not try another, because I choose to use the Learning board to do power, need to connect to the computer power supply, so there will be COM3 and COM4, where COM3 is the learning Board of the USB interface. After opening the serial port, enter D is to measure the distance, input o can see infrared light.

Where the left is the measuring distance, and the right is the millisecond

Of course, you can also use software, directly write programs, personal programs written as follows:

For reference only, the specific implementation varies from person to person.

1. #include <reg51.h>
2. #define uchar unsigned char
3. #define uint unsigned int
4. Sbit led=P1^7; //ok
5. Sbit k2=P2^5;
6. Sbit k3=P2^6;
7. Sbit k4=P2^7;
8. Uint Sysec;
9. Uchar ErrorCode;
10. Uchar stringLenth;//(Serial port received string length, single measurement received 28 bytes continuous Measurement 38 bytes)
11. #define maxSbufLenth 38 //(Set cache size based on measurement mode)
12. Uchar aciiCount; //(Calculate the number of characters received)
13. Uchar multipleTestShift; //(Continuous measurement to subtract an offset of 10 bytes)
14. Uchar uartSbuf[38]={"$00023335&$0006210000000000&"};//(The 20th bit is the distance of 10 digits 21 is the single digit, followed by the decimal)
15. Uchar xdata laserOn[12] ={"$0003260130&"};
16. Uchar xdata singleTest[10] ={"$00022123&"};
17. Uchar xdata multipleTest[10] ={"$00022426&"};
18. Void Timer0Init(void) //50ms@6.000MHz
19. {
20. AUXR &= 0x7F; //Timer clock 12T mode
21. TMOD &= 0xF0; //Set the timer mode
22. TMOD |= 0x01; //Set the timer mode
23. TL0 = 0x58; //Set the initial value of the timing
24. TH0 = 0x9E; //Set the initial value of the timing
25. TF0 = 0; //Clear the TF0 flag  
    //
26. TR0 = 1; //Timer 0 starts timing
27. ET0=0;
28. EA=1;
29. }
30. Void UartInit(void) //115200bps@11.0592MHz
31. {
32. PCON &= 0x7F; //The baud rate is not doubled
33. SCON = 0x50; //8-bit data, variable baud rate
34. AUXR |= 0x40; //Timer 1 clock is Fosc, ie 1T
35. AUXR &= 0xFE; //Serial 1 selects timer 1 as the baud rate generator
36. TMOD &= 0x0F; //Clear the timer 1 mode bit
37. TMOD |= 0x20; //Set timer 1 to 8-bit auto reload mode
38. TL1 = 0xFD; //Set the initial value of the timing
39. TH1 = 0xFD; //Set the timer reload value
40. ET1 = 0; // disable timer 1 interrupt
41. TR1 = 1; //Start timer 1
42. ES=1;
43. EA=1;
44. }  
    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*Send character (ASCII) function \*\*\*\*\*\*\*\*\*\*\*\*\*/
45. Void sendAscii(uchar \*b)
46. {
47. ES = 0; //Off serial port interrupt
48. For (b; \*b!='\0';b++)
49. {
50. SBUF = \*b;
51. While (TI!=1); //waiting to send complete
52. TI = 0; //Clear the transmit interrupt flag
53. }
54. ES = 1; //Open serial port interrupt
55. }
56. Void clearUartSbuf()
57. {
58. Uchar i;
59. For (i=0;i<aciiCount;i++)
60. {
61. uartSbuf[i]='0';
62. }
63. }
64. Void main()
65. {
66. Uchar test;
67. UartInit();
68. While (1)
69. {
70. If (k4==0)
71. {
72. While (k4==0);
73. sendAscii("$0003260130&");//Open the laser
74. }
75. If (k3==0){
76. While (k3==0);
77. ErrorCode=0;
78. stringLenth=28;
79. aciiCount=0;// receiving 38 strings is enough
80. sendAscii("$00022123&");//Single measurement or turn off

continuous measurement

}  
If (k2==0){  
While (k2==0);  
stringLenth=38;  
aciiCount=0; / / receiving 28 strings is enough  
sendAscii("$00022426&");//Start continuous measurement  
}  
}  
}  
Void uartRec()interrupt 4{  
ES=0; / / close interrupt  
If (RI)  
{  
RI=0;  
uartSbuf[aciiCount]=SBUF;  
aciiCount++;  
If (aciiCount>stringLenth) //(Intercept strings of the right length based on each pattern)  
{  
aciiCount=0;  
}  
}  
If (TI)  
{  
TI=0;  
}  
ES=1;  
}  
Void t0() interrupt 1  
{  
Static char i;  
TL0 = 0x58; //Set the initial value of the timing  
TH0 = 0x9E; //Set the initial value of the timing  
i++;  
If (i==20)  
{  
i=0;  
Sysec++;  
}  
}