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.

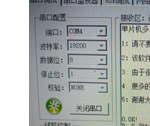
Just four wires, after the welding is complete. The 3V power connection is as the picture.

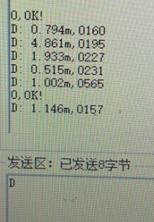
follows

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.

#include <reg51.h>

#define uchar unsigned char

#define uint unsigned int

sbit led=P1^7; //ok

sbit k2=P2^5;

sbit k3=P2^6;

sbit k4=P2^7;

uint Sysec;

uchar ErrorCode;

uchar stringLenth;//(Serial port received string length, single measurement received 28 bytes continuous Measurement 38 bytes)

#define maxSbufLenth 38 //(Set cache size based on measurement mode)

uchar aciiCount; //(Calculate the number of characters received)

uchar multipleTestShift; //(Continuous measurement to subtract an offset of 10 bytes)

uchar uartSbuf[38]={"$00023335&$0006210000000000&"};//(The 20th bit is the distance of 10 digits 21 is the single digit, followed by the decimal)

uchar xdata laserOn[12] ={"$0003260130&"};

uchar xdata singleTest[10] ={"$00022123&"};

uchar xdata multipleTest[10] ={"$00022426&"};

void Timer0Init(void) //50毫秒@6.000MHz

{

AUXR &= 0x7F; //定时器时钟12T模式

TMOD &= 0xF0; //设置定时器模式

TMOD |= 0x01; //设置定时器模式

TL0 = 0x58; //设置定时初值

TH0 = 0x9E; //设置定时初值

TF0 = 0; //清除TF0标志

//TR0 = 1; //定时器0开始计时

ET0=0;

EA=1;

}

void UartInit(void) //115200bps@11.0592MHz

{

PCON &= 0x7F; //波特率不倍速

SCON = 0x50; //8位数据,可变波特率

AUXR |= 0x40; //定时器1时钟为Fosc,即1T

AUXR &= 0xFE; //串口1选择定时器1为波特率发生器

TMOD &= 0x0F; //清除定时器1模式位

TMOD |= 0x20; //设定定时器1为8位自动重装方式

TL1 = 0xFD; //设定定时初值

TH1 = 0xFD; //设定定时器重装值

ET1 = 0; //禁止定时器1中断

TR1 = 1; //启动定时器1

ES=1;

EA=1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*发送字符(ASCII)函数\*\*\*\*\*\*\*\*\*\*\*\*\*/

void sendAscii(uchar \*b)

{

ES = 0; //关串口中断

for (b; \*b!='\0';b++)

{

SBUF = \*b;

while (TI!=1); //等待发送完成

TI = 0; //清除发送中断标志位

}

ES = 1; //开串口中断

}

void clearUartSbuf()

{

uchar i;

for (i=0;i<aciiCount;i++)

{

uartSbuf[i]='0';

}

}

void main()

{

uchar test;

UartInit();

while (1)

{

if (k4==0)

{

while (k4==0);

sendAscii("$0003260130&");//打开激光

}

if (k3==0){

while (k3==0);

ErrorCode=0;

stringLenth=28;

aciiCount=0;//接收38个字符串就够了

sendAscii("$00022123&");//单次测量 或者 关闭连续测量

}

if (k2==0){

while (k2==0);

stringLenth=38;

aciiCount=0;//接收28个字符串就够了

sendAscii("$00022426&");//开始连续测量

}

}

}

void uartRec()interrupt 4{

ES=0;//关闭中断

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; //设置定时初值

TH0 = 0x9E; //设置定时初值

i++;

if (i==20)

{

i=0;

Sysec++;

}

}