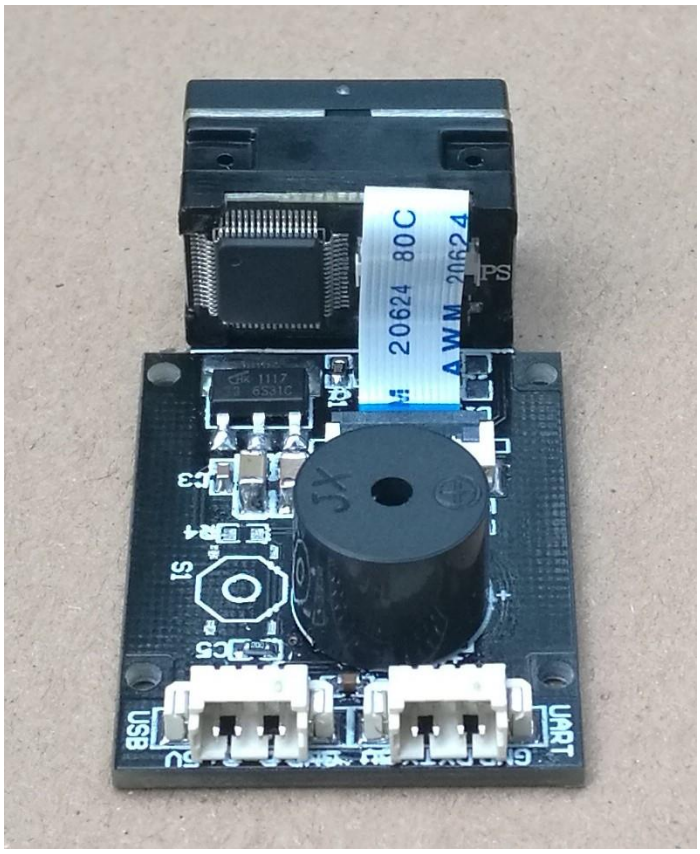
TIPS:本教程适用于 **arduino uno** 开发板使用城章科技的 **GM65** 模块进行串口通信，其他开发板也可参照本例程使用 **GM65** 模块

一、GM65 模块初始设置



我们可以看到模块的尾部有两个接口：USB&UART，usb 可以用作模块和电脑串口通信、模块作为 usb 虚拟键盘使用（当模块接在 pc 且设置为标准 usb 键盘输出的时候，电脑的文字输入区会显示模块扫描二维码的信息）；UART 接口用于真实串口通信，也是今天要讲的内容。

店家随模块附加了两个文件

GM65 串口简要设置.docx
城章科技 GM65条码识别模块用户手册.pdf

附件一：GM65 串口简要设置

附件二：用户手册

附件二里有一堆二维码，用模块的摄像头扫一下就能进行相应的设置，每个二维码对应有不同的设置选项，我们先来看一下模块的默认设置

默认识读模式		自动模式	
单次读码时间		3s	参数范围：0.1-25.5 秒，步长为 0.1s；0 表示单次解码时间不限
读码间隔		1S	参数范围：0.1-25.5 秒，步长为 0.1s；0 表示单次解码时间不限
输出编码		GBK 编码输出	GBK 编码、UNICODE 格式、 BIG5 格式
接口方式		标准 USB 键盘输出	USB 键盘输出、串口输出、USB 虚拟串口输出
当使用 TTL-232 接口时	波特率	9600	波特率可自行设置，详见 2.1 节
	校检	无校检	
	数据位	8 位	
	停止位	1 位	
	硬件流控	无硬件流控	
串口触发模式	单次读码时间	5s	参数范围：0.1-25.5 秒，步长为 0.1s；0 表示单次解码时间不限

实际上我们不需要对这些参数做过多调整，把接口方式改成串口输出直接用就行

←串口输出设置码

但是使用的时候还会遇到两个问题：

- 模块灯光太亮会导致呈现二维码的手机屏幕或者电脑屏幕反光，无法准确识别二维码
- 读的时候响声会很烦人

这里建议把灯光

××××××××××××××××××××××××××××××××××××

把两个二维码隔开效果更佳

把读取音设置为

第一次使用的时候可以先把模块接 usb 口插电脑设置为标准键盘输出，然后在 https://cli.im/生成一个二维码，打开一个 word，然后用模块读取，看看是不是你预先设置好的信息。

如果没问题，那么请把模块设置为串口输出，无光和静音随意。

拆掉 usb 线，把附带的排线接到 UART 口，开始下一步。

二、模块与 arduino uno 板子的连接

GM65 → UNO

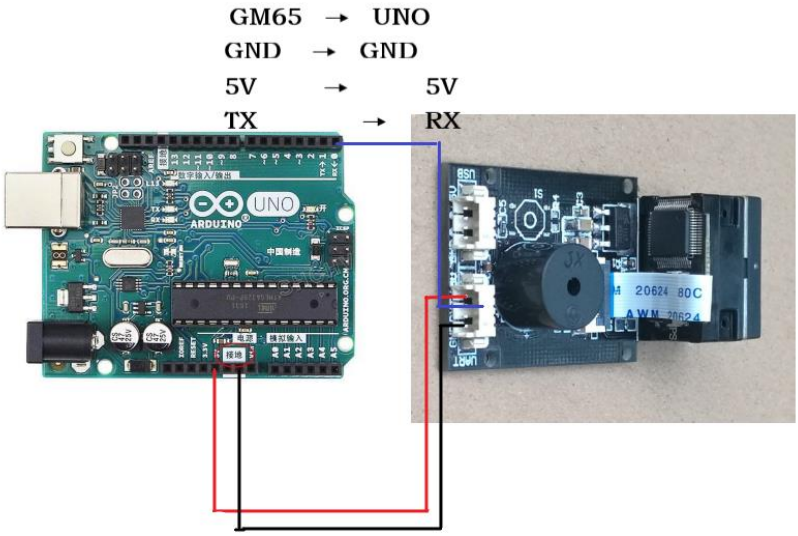
GND → GND

5V → 5V

TX → RX

进行串口通信需要两根数据传输线，一根为 RX(Receive 接收)，一根为 TX(Transmit)，在进行双向通信时两个设备的异名端口相互连接

需要注意的是,我们这次用到的是 GM65 发送、UNO 接收的单向通信，所以只需要将 GM65 的 TX 口和 UNO 的 RX 口接到一起，以及 5V、GND 对应连接



三、arduino 通过串口读取模块识别的信息

```
char info;
void setup() {
Serial.begin(9600,SERIAL_8N1); //设置串口波特率9600

}

void loop() {
  info= Serial.read();//Serial.read()串口读取为串口读取函数
  Serial.println(info);
  delay(100);

}
```

还记得之前的默认波特率参数吗，如果板子的波特率和模块的波特率不相同是无法通信的噢

代码第一行声明了一个字符串 info

代码 set up（）函数为 UNO 的初始设置，这里我们暂时只需要让板子的波特率和串口的波特率相匹配

代码的 loop 函数会在程序被烧录到板子上后循环执行，执行间隔由 delay 函数确定；Serial.read()函数为串口读取函数；Serial.print()和 Serial.print()为串口输出函数。

注意：烧录的时候需要把 UNO 和 GM65 的数据传输线断开，因为烧录这个过程也是串口通信--由 PC 向 UNO 发送数据，如果不断开的话 MG65 也在持续向板子发送数据，也就是说串口被 GM65 占用了，因此程序大概率无法烧录到板子上

依旧是在 https://cli.im/生成一个二维码（建议纯数字），我们以 1234 为例

首先将程序烧录到然后打开串口助手，你会惊奇的串口助手显示的是这个样子？？？？？其实是上面的代码少了一步逻辑判断

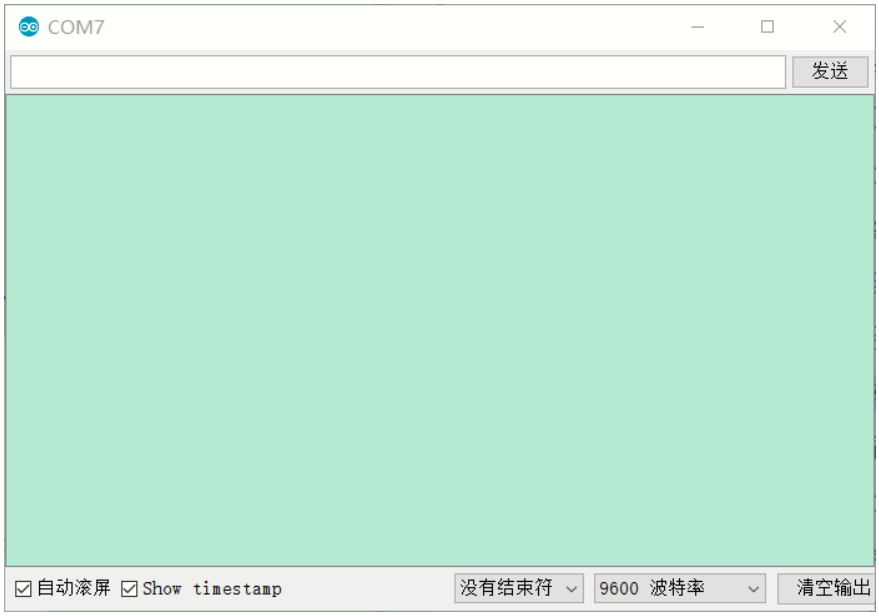
```
15:14:05.034 -> ?
15:14:05.138 -> ?
15:14:05.241 -> ?
15:14:05.344 -> ?
15:14:05.447 -> ?
15:14:05.549 -> ?
15:14:05.653 -> ?
15:14:05.755 -> ?
15:14:05.858 -> ?
15:14:05.962 -> ?
15:14:06.066 -> ?
15:14:06.169 -> ?
15:14:06.272 -> ?
15:14:06.376 -> ?
15:14:06.446 -> ?
15:14:06.550 -> ?
15:14:06.652 -> ?
15:14:06.756 -> ?
```

我们简单的修改一下代码，在 loop 内设置一个判断

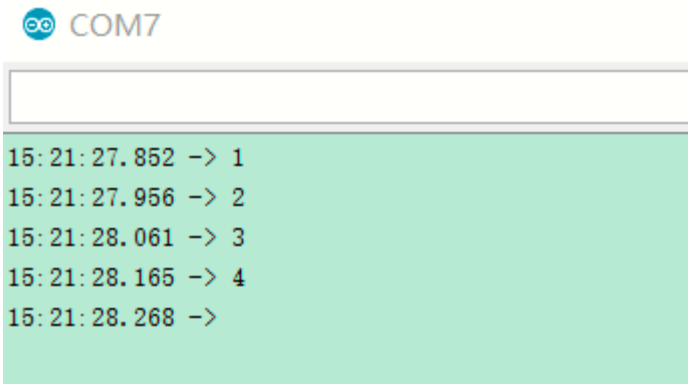
```
char info;
void setup() {
Serial.begin(9600,SERIAL_8N1); //设置串口波特率9600

}

void loop() {
  if (Serial.available() > 0){
  info= Serial.read();//Serial.read()串口读取为串口读取函数
  Serial.println(info);
  delay(100);

}
}
```

Serial.available()>0 说明串口接收到了数据，如果接受不到就不接受数据。

再次烧录后打开串口助手。



然后我们扫描上面的一个二维码就会发现信息被读出来了



四、如何通过程序处理 GM65 读取的信息

上面的图可以看出来我们设置的二维码的信息确实是被串口读取了，但是四位数字被拆分成了一位一位输出，这对我们对信息的提取和处理造成了很大麻烦。不妨在 UNO 的程序上设置一些函数帮我们处理这些问题，我们依旧是拿数字二维码做文章，这一次，我们设置一个内容为：ssss178884888481234 的二维码

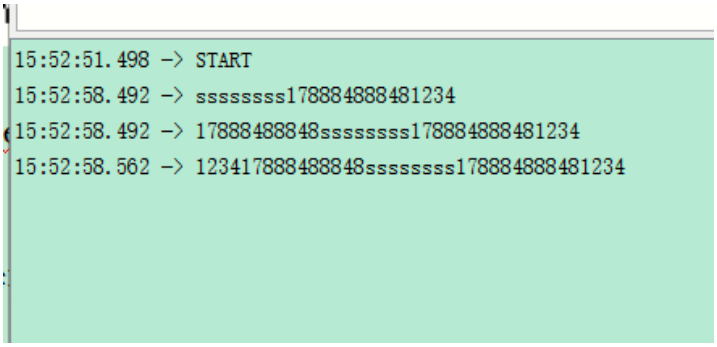
这次我们将 processtset 的代码烧到 UNO

代码在下一页

这份代码我觉得就不需要过多解释了，没有什么需要注意的

然后我们打开串口助手，扫描上面的二维码，就会得到我们不想要的结果：

（注：为了方便程序调试，我们在 setup 函数内设置一个开始标志输出 START）



我们需要输出的是 id 内包含的内容，但是实际上输出的却是 id+temp
我们需要输出的是 nmb 内包含的内容，但是实际上输出的却是 nmb+id+temp
我们需要输出的是 food 内包含的内容，但是实际上输出的却是 food+nmb+id+temp

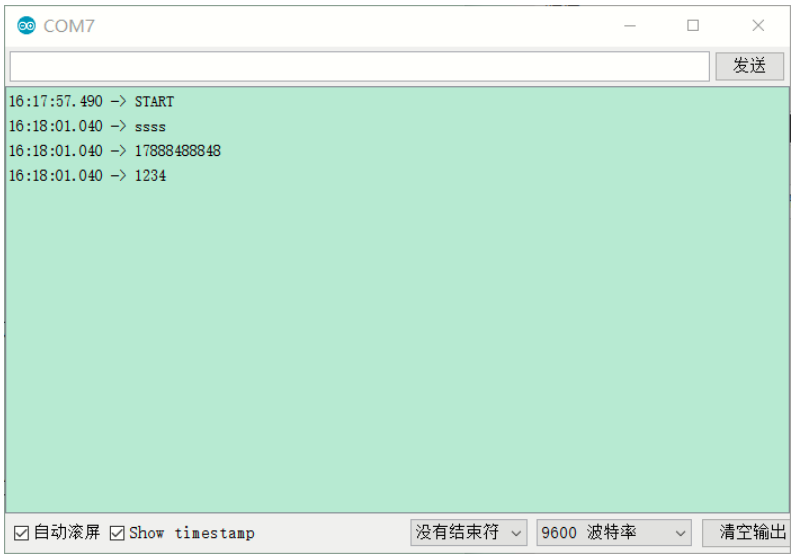
```
#include<string.h>
#define ID 4
#define NMB 11
#define FOOD 5
char info;
char temp[20];
char id[ID];
char nmb[NMB];
char food[FOOD];
int i=0;
```

```
void setup() {
  Serial.begin(9600,SERIAL_8N1); //设置串口波特率9600
  Serial.println("START");
  }
 void loop() {
  if (Serial.available() > 0){
      i++;
      info= Serial.read();
      temp[i-1]=(char)info;
      if(i==20){
          divID(temp);
          divNMB(temp);
          divFOOD(temp);
          Serial.println(id);
          Serial.println(nmb);
          Serial.println(food);
      }
   }
   delay(100);
}
```

```
void divID(char a[]){
  for(int j=0;j<ID;j++){
    id[j]=a[j];
  }
}
void divNMB(char a[]){
  for(int j=0;j<NMB;j++){
   nmb[j]=a[j+ID];
  }
}
void divFOOD(char a[]){
  for(int j=0;j<FOOD;j++){
    food[j]=a[j+ID+NMB];
  }
}
```

图三为用于切割 temp 数组的三个函数

想要解决输出异常其实很简单，只需要把要输出的数组后面附加的数组置零就可以



修改的部分代码如下

```
void loop() {
 if (Serial.available() > 0){
     i++;
     info= Serial.read();
     temp[i-1]=(char)info;
     if(i==20){
         divID(temp);
         divNMB(temp);
         divFOOD(temp);
       memset(temp,0,21);
         Serial.println(id);
       memset(id,0,ID);
         Serial.println(nmb);
       memset(nmb,0,NMB);
         Serial.println(food);
     }
  }
  delay(100);
}
```

Memset()函数为数组清零函数

友情提供 GM65 二维码模块淘宝购买链接：

https://item.taobao.com/item.htm?id=552884247068