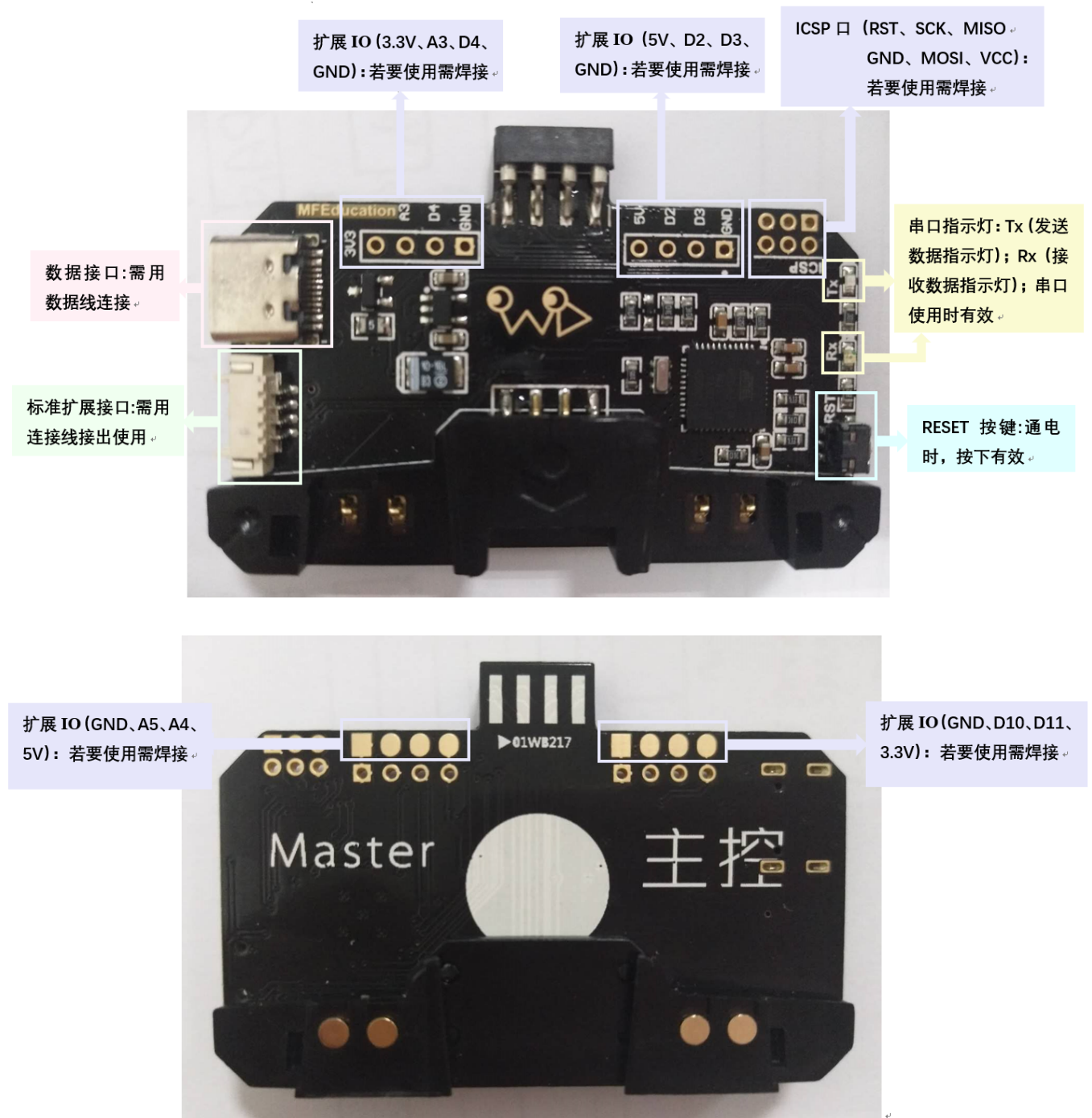


# 主控-Master

## 模块介绍

主控模块与我们人类的大脑、电脑的CPU的作用类似，在豌豆拼夸克中，它是指挥 和控制 其它模块的中心，也能与其它模块进行通信。主控是所有设计中必须使用的模块。



类别	详细介绍	基本示例	扩展示例
----	------	------	------

数据接口	USB口，用配件中的 <b>数据线</b> 将主控模块连接到电脑后，可以供电、下载程序及串口通信	无	无
标准扩展接口	一个可与白色模块连接的标准接口	<b>LED彩灯使用</b>	持续更新
串口指示灯	Tx（发送数据指示灯），Rx（接收数据指示灯），通过串口传递数据时，串口指示灯亮	<b>串口使用</b>	<b>温度报警器</b>
<b>RESET按键</b>	具有复位和清空程序的功能	无	无
<b>扩展IO</b>	3.3V、A3、D4、GND	持续更新	持续更新
	5V、D2、D3、GND		
	GND、A5、A4、5V		
	GND、D10、D11、3.3V		
<b>ICSP口</b>	硬件SPI通信接口，很少用到	持续更新	持续更新

## 使用示例

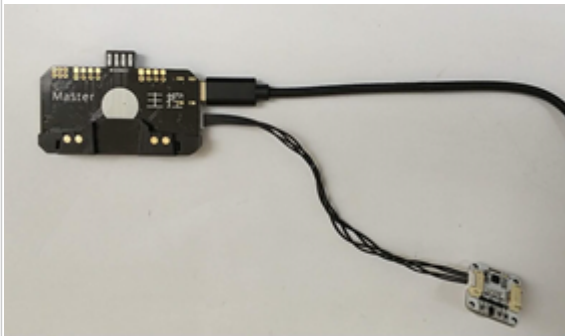
### 基本示例

#### 一. LED彩灯使用

##### 1.功能说明和硬件连接

**清单：**主控模块，LED彩灯白色模块、连接线

**功能：**程序下载成功后，LED彩灯点亮。

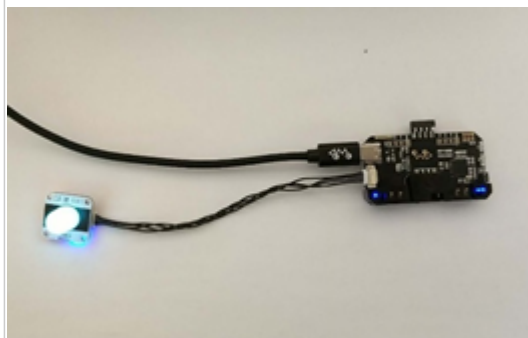


##### 2.软件代码

```
/*  
  
* 通过主控板标准扩展接口点亮RGB灯  
  
*/  
  
void loop() {  
  
    LED1.setRGB(10, 20, 30); /*控制彩灯RGB值（红：0~255，绿：0~255，蓝0~255）*/  
  
}
```

}

### 3.实现图片



## 二. 串口使用

### 1.功能说明和硬件连接

**清单：** 主控模块、显示模块

**功能：**

程序下载成功后，点击豌豆拼IDE右上角的串口，在发送输入框中输入1后点击发送，Rx指示灯闪烁一下黄灯，接收框中隔1s打印“Hello,WonderBits”，Tx指示灯闪烁绿灯，同时显示模块屏幕中显示1；在串口的发送输入框中输入除1之外的任何数，Rx指示灯亮闪烁一下黄灯，串口接收框中停止显示，同时显示模块屏幕中显示该数。



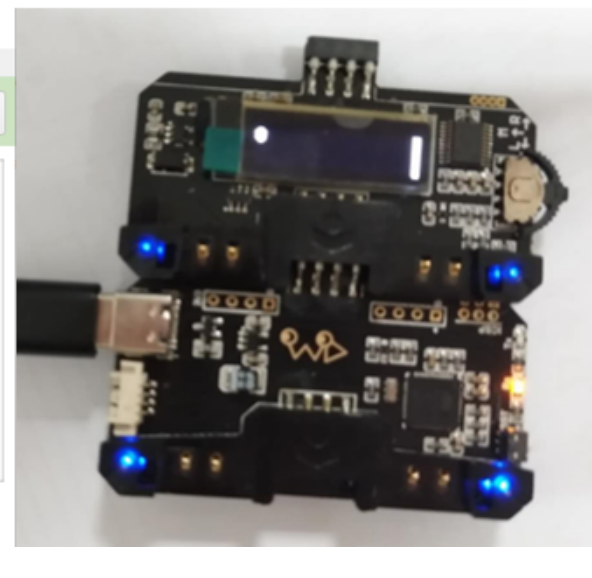
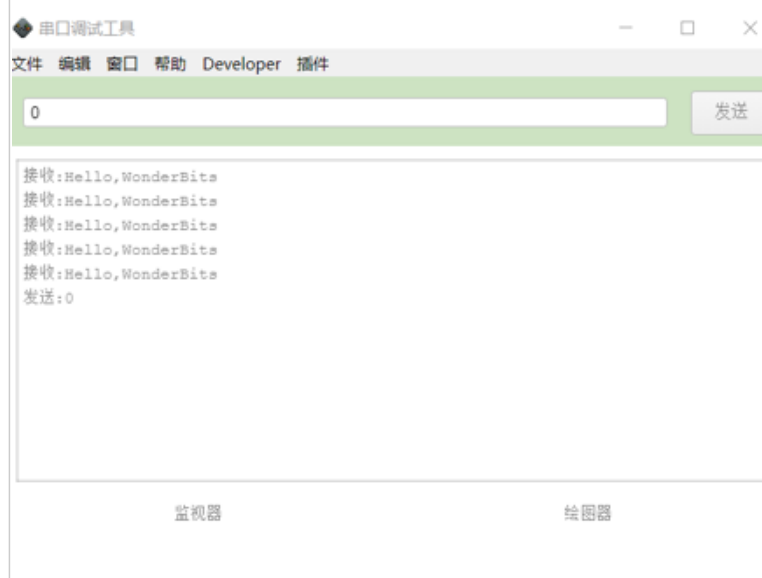
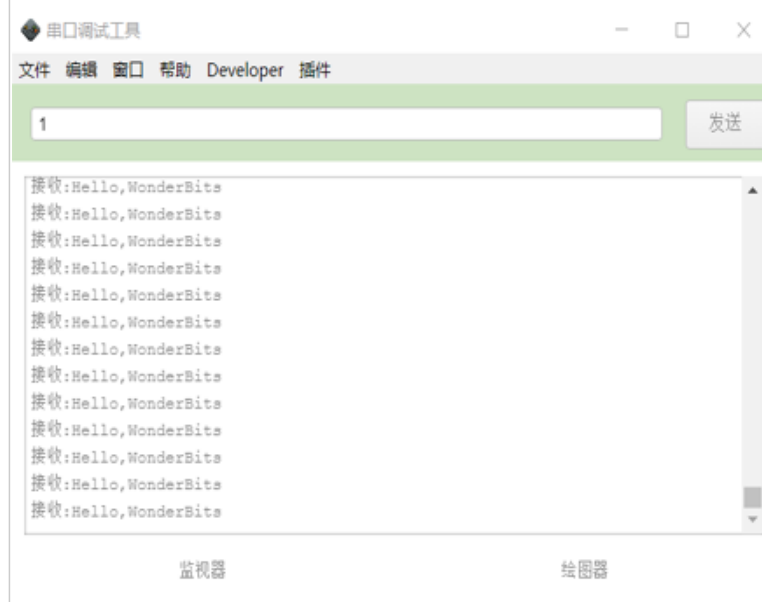
### 2.软件代码

```
/*  
  
* 串口通信  
*/  
void loop()  
{  
  if (sysmaster.serial.getValue() == 1)  
    // 如果串口发送的数为1，则串口打印Hello,WonderBits  
    {  
      sysmaster.serial.println("Hello,WonderBits");  
    }  
}
```

```
Display1.print(1, 1, sysmaster.serial.getValue());  
// 将串口发送来的数显示在显示模块的屏幕中  
  
delay(1000);  
}
```

### 3.实现图片

(1) 串口发送1, 串口打印信息, 显示屏显示1。(2) 串口发送0 (或1以外的任何数), 串口停止打印信息, 显示屏显示该数。



## 扩展示例

### 一、温度报警器

#### 1.功能说明和硬件连接

**清单：** 主控模块、显示模块、监测模块和信号模块

**功能：**

程序下载成功后，在串口调试工具中输入温度上限的值，监测模块中的温度传感器实时监测并显示当前所处环境的温度，若高于温度上限，则报警，以实现温度报警器的功能。具体步骤及效果见“

3.步骤及实现图片”。



## 2.软件代码

```
/*  
 * 温度报警器  
*/  
  
int TempHigh, Nowtemp;  
  
void setup()  
{  
    Display1.print(1, 1, "温度上限:");  
    Display1.print(2, 1, "当前温度:");  
    delay(500);  
  
    sysmaster.serial.println("请输入温度上限:");  
  
    while(sysmaster.serial.getValue()==0);  
  
    TempHigh=sysmaster.serial.getValue();  
  
    Display1.print(1, 10, TempHigh);  
}  
  
void loop()  
{
```

```
Nowtemp=Observer1.getTemperature();  
Display1.print(2, 10, Nowtemp);  
sysmaster.serial.println(Nowtemp);  
if (Nowtemp >TempHigh)  
{  
Signal1.setBuzzer(800);  
}  
else  
{  
Signal1.setBuzzer(0);  
}  
delay(500);  
}
```

### 3.步骤及实现图片

(1)

程序下载成功后，打开串口调试工具，接收框显示提示信息“请输入温度上限：”。（若没有接收到提示信息，可按下主控模块中的**RST**按键 复位）

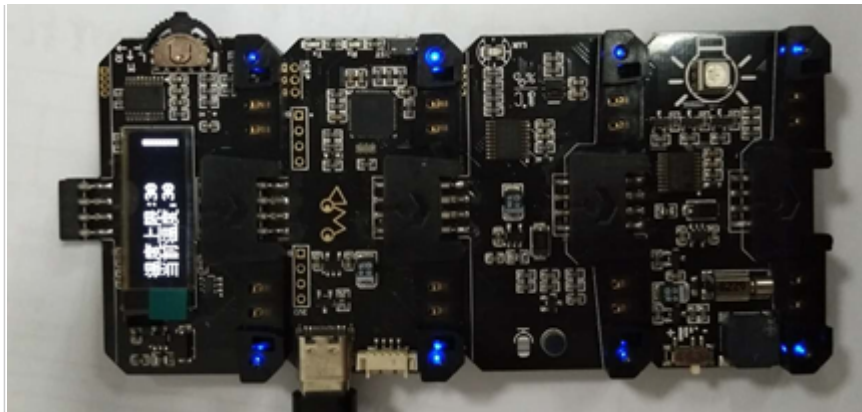


(2) 在发送输入框内输入温度上限的值，比如30，点击发送。



(3) 此时监测模块中的温度传感器获取当前环境温度，并将其显示在显示模块的OLED显示屏及串口调试工具的接收框中。一旦当前环境温度大于设置的上限，信号模块中的蜂鸣器开始报警。(本示例测得的当前温度为30，使用者在不同的环境中会测得不同的当前温度，结果不一定跟本示例相同)





## 常见问题

**问：**程序下载失败怎么办

**答：**待添加

**问：**串口指示灯亮灭是什么意思？

**答：**Tx（发送数据指示灯）亮一下绿灯 表示串口发送数据一次，Rx（接收数据指示灯）亮一下黄灯 表示串口接收数据一次。串口发送数据指的是主控模块给电脑的串口调试工具发送数据，同理可得串口接收数据的意义。

**问：**怎么知道串口是否在正常通信？

**答：**针对基本示例（二.[串口使用](#)），首先检查是否打开了串口窗口，即点击豌豆拼IDE右上角串口



串口

后出现的窗口，然后在发送输入框

1

发送

输入1之后点击发送，看主控模块中的Rx指示灯是否亮了一下黄灯，最后接收框出现打印信息且主控模块的Tx指示灯闪烁绿灯。若出现上述的所有情形，说明串口正在正常通信。





## 原理介绍

### • RESET按键

1. 复位功能：通电状态，按住然后松开，即可让主控重新开始工作；
2. 清空功能：拔掉USB断电状态，按住主控复位按键连接USB上电三秒后，主控板上的绿灯亮起即主控模块已经执行出场设置，可用于清空主控板程序。

### • 扩展IO

主控模块中共有四组扩展IO，其中3.3V、5V代表电压为3.3伏和5伏，GND代表地，其中A开头的为模拟IO口，D开头的为数字IO口，当外接非模块化器件时可以使用，扩展IO和Arduino上对应编号的针口类型相同。

### • ICSP 口

ICSP口为在线串行编程接口，因为[arduino](#)上面配了32U4等USB控制器，可以通过USB口通信烧写程序。若要使用该口下载程序，首先需焊接专用接口，其次需使用专用下载器。ICSP一般很少用到。

### • 主控模块单片机

主控模块是基于 Atmel ATmega32u4开发的单片机核心板，软件上兼容[Arduino Leonardo](#) 的所有函数。

ATmega32U4芯片具有以下功能特点：


1. 具有最先进的RISC（精简指令集）构架：135条强大的指令，32\*8个通用工作寄存器，全静态操作，在16MHz时钟频率下高达16MIPS的吞吐量，两周期循环芯片；
2. 不易失的程序和数据存储器（SRAM）：内置32KB可编程闪存，12.5KB的内部静态随机存取存储器（SRAM），1KB的内部电可擦可编程只读存储器（EEPROM），写入/擦除周期10000Flash/100000EEPROM，数据保存时间为85°C下20年/25°C下100年，具有编程闪光灯和EEPROM的安全锁

3. 其他特点：一个8位定时器/计数器预定标器和两个PWM频道，4通道10-bit ADC，可编程看门狗定时器和单独的振荡器，模拟比较器；
4. 特殊的微控制器功能：debugWIRE在芯片调试系统，SPI通讯端口，外部和内部中断源，低功耗空闲，ADC噪声抑制，掉电模式，增强的上电复位电路，具有软件禁用功能的可编程输出检测电路，内部校准振荡器；
5. I/O和程序包：8引脚PDIP/SOIC（6个可编程I/O口），10-pad MLF（6个可编程I/O口），20-pad MLF（6个可编程I/O口）；
6. 工作电压：1.8-5.5V
7. 速度等级：0-4MHz@1.8-5.5V, 0-10MHz@2.7-5.5V, 0-20MHz@4.5-5.5V;
8. 低功耗：主动模式下190μA/1.8V/1MHz，空闲模式下24μA/1.8V/1MHz。

From:

<http://wiki.wonderbits.cc/> - 豌豆拼Wiki

Permanent link:

<http://wiki.wonderbits.cc/doku.php?id=%E6%A8%A1%E5%9D%97:%E4%B8%BB%E6%8E%A7> 

Last update: **2018/08/25 08:57**