

目录

一、米思齐简介.....	2
二、 米思齐使用手册.....	2
1. 软件界面介绍.....	2
2. 各功能模块介绍.....	3

一、米思齐简介

Mixly（全称 Mixly_Arduino，中文名称米思齐）是一款将图形化编程方式和代码编程方式融合在一起，为硬件编程的软件开发环境，是在北京师范大学教育学部创客教育实验室负责人傅骞老师的带领下，由其团队开发的一款基于 Blockly 和 Java8 自主研发，且免费开源的图形化编程工具。

目前，开源硬件 Arduino 中的 AVR 系列均可通过米思齐来开发。与 Arduino 的可视化编程插件 ArduBlock 相比，米思齐简化了 Arduino IDE 和 ArduBlock 可视化编程插件的双窗口界面，为 Arduino 学习者提供了更友好的编程环境。

二、米思齐使用手册

1. 软件界面介绍

米思齐软件界面如图 1.1 所示。



图 1.1 米思齐的软件界面

总体来说，软件界面分为四个部分：

（1）左侧为模块区，这里包含了米思齐中所有能用到的程序模块，根据功能的不同们大概分为以下几类：输入/输出、结构控制、数学变换、文本输出、数组列表、逻辑处理、传感模块、执行模块、通讯模块、显示模块、变量常量、函数处理。

（2）模块区的右侧是程序构建区，按住鼠标左键拖住模块区的模块，可以将它们放到程序构建区，拖拽过来的模块会在这里组合成一段有一定逻辑关系的程序块。在这个区域的右下角有一个垃圾桶，想要删除模块时，就要将模块拖入垃圾桶中。在垃圾桶上方的三个圆形按钮，能够实现程序块的放大、缩小和居中。

(3) 模块区和程序构建区的下方是基本功能区，有点类似一般软件的菜单区。这里不仅包含了新建、打开、保存、另存为这样的基本按键，还包含了硬件编程软件中需要用到的编译、上传、控制板选择、连接端口选择及串口监视器这样的按钮。

(4) 软件最下方是提示区，这里在软件编译、上传过程中会显示相应的提示信息。我们可以通过提示信息来解决编译上传中出现的一些问题。

(5) 除以上四个基本模块外，米思齐还支持多语言选择功能，可以通过界面右上角的下拉菜单里选择不同的语言版本。除此之外米思齐界面左上角模块的右侧有一个代码选项卡，单机选项卡能进入纯代码形式。米思齐是一款将图形化编程方式和代码编程方式融合在一起的开发环境，此时（如图 1.2 所示），在程序构建区的右侧就会显示出相应的代码，会随着模块的变化而变化。



图 1.2 代码程序和图形程序同时呈现

2. 各功能模块介绍

2.1 程序模块区

2.1.1 输入/输出

输入输出所包含的指令主要分为四部分：控制管脚的输入输出（按信号类型可分为数字信号和模拟信号）、中断、脉冲长度及 ShiftOut。

输入输出：数字输入、数字输出、模拟输入、模拟输出、中断控制、脉冲长度、移位输出。



图 2-1-1-1 输入输出指令模块

2.1.1.1 输入/输出——数字/模拟输入/输出



图 2-1-1-2 数字/模拟输入/输出

依据信号是数字信号还是模拟信号以及信号的状态是输入还是输出进行适当的物理连接及代码选择。例如：如果想要点亮一盏 LED 灯（假设该灯连接管脚 13），使用

数字输出语句将管脚 13 的电平调为高即可。

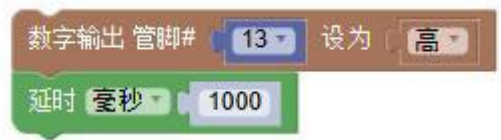


图 2-1-1-3 点亮一盏 LED 灯的代码

如果想要使用模拟信号，就需要用到模拟管脚，当学生用模拟输出或模拟输入指令时，单击其下拉箭头，可看到 Arduino 所支持的模拟输出或模拟输入所支持的管脚口，省去学生记忆的时间。



图 2-1-1-4 设置模拟输出管脚



图 2-1-1-5 设置模拟输入管脚

2.1.1.2 输入/输出——中断指令



图 2-1-1-6 中断指令模块

由于程序往往是顺序执行，但有时可能要突然打断正在执行的指令而转而去执行更加重要的指令，因此，中断的使用就显得尤为重要了。

以下是一个最典型的中断指令应用的案例，即按一下开关按键 LED 点亮三秒，三秒后自动熄灭，但如果在灯亮三秒内监测到开关按键再次按下，则 LED 立即熄灭。



图 2-1-1-7 中断指令应用案例



图 2-1-1-8 设置中断模式

Mixly 的中断模式有三种可以选择：上升、下降及改变。

上升指的是按键的电平由低到高（实际上就是按键按下的过程）；下降指的是按键的电平由高到低（实际上就是按键抬起的过程）；

改变指的是按键的电平由低到高或由高到低的过程（实际上就是按键按下或抬起的过程）。

当学生使用中断上升模式时，只要按下按键则立即执行中断语句内的指令，当执行完中断内指令后，代码将继续执行之前被打断的相应指令。

2.1.1.3 输入/输出——脉冲长度



2-1-1-9 输入/输出中的脉冲长度

2.1.1.4 输入/输出——ShiftOut



图 2-1-1-10 输入/输出中的 ShiftOut

2.1.2 控制

时间延迟、条件执行、循环执行、获取时间、初始化、Switch 执行（新增）控制模块中主要执行的内容是对程序结构进行的相应控制。



图 2-1-2-1 控制指令模块

2.1.2.1 控制——初始化

初始化中的内容程序只执行一次，因此我们通常把一些变量的声明（如下图 2-2-3）或函数定义（如中断）等放在初始化里。



图 2-1-2-2 初始化一个整型变量

图 2-1-2-3 对中断模式进行初始化

2.1.2.2 控制——条件语句

Mixly 的条件语句为两个，一个是 ，另一个是 。

以  为例：



图 2-1-2-4 条件语句代码示例

可以理解为：当按键按下（按键接管脚 2）时（此时【数字输入管脚#2】状态为 1，满足条件句执行条件），LED（接管脚 13）点亮；否则（即按键抬起后，【数字输入 管脚#2】状态为 0，不满足执行条件，自动执行否则里面的内容）LED 灭。在使用条件语句过程中需注意，该条件句左上角有一个小齿轮，当点击小齿轮时可得到图 2-1-15，把否则拉入条件句内，如下图 2-1-16 所示。当对条件句操作完后，需再次点击左上角的小齿轮将弹出的窗口关闭。



图 2-1-2-5 点击小齿轮展开“如果”模块

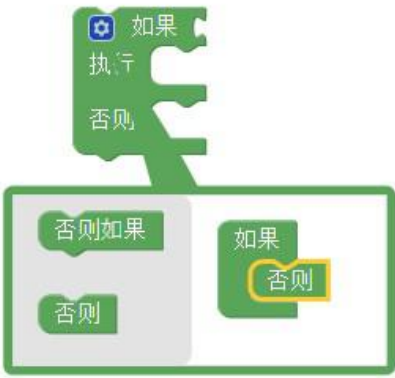


图 2-1-2-6 设置“如果-否则”语句

2.1.2.3 控制——循环语句

控制循环语句的代码有三条。

分别是  、  。


循环语句的功能就是使某段代码重复运行。比如最常见的呼吸灯（手机提醒灯或者路由器的灯）。呼吸灯的效果是 LED 由慢慢由灭变亮，后又慢慢由亮变灭的一种渐变灯。对于存在中间状态的渐变，用数字输出就无法实现了。



图 2-1-2-7 呼吸灯的代码示例

图 2-1-2-7 就是呼吸灯的代码。Arduino 本身支持的模拟输出状态为 256 种，也就是 0——255 个数值。步长为每次循环变量增加的幅度。“使用 i 从 0 到 255 步长为 1”可以理解为 i 将以每次加 1 的增幅由 0 变化到 255。而我们在 3 号管脚接的 LED 的值也将是 0, 1, 2, 3……最后直到 255。第二个循环块原理类似。

2.1.2.4 控制——系统运行时间

 该指令可以读取从代码编程开始到此时的系统时间。

2.1.3 数学

该部分主要完成数学变换功能，具体包括数学运算、三角函数、随机、取整、取最大值、数字约束、数字映射。



图 2-1-3-1 数学指令模块

2.1.3.1 数学——随机数



图 2-1-3-2 生成随机数模块

随机数是一个左闭右开的取值范围。也就是较小的数可以取到，较大的数无法取到。

2.1.3.2 数学——约束



图 2-1-3-3 约束模块

约束是将数值限制在最小值与最大值之间，所有小于最小值的数值被赋值为最小值，同理，所有大于最大值的数被赋值为最大值。

2.1.3.3 数字——映射



图 2-1-3-4 映射模块



图 2-1-3-5 LED 与蜂鸣器同步增减

图 2-1-3-5 为 LED 与蜂鸣器同步增大或减小的代码。因为对于 $\sin(i)$ 来说，其值域为正数时 i 的取值范围为 $[0, 180]$ ，不过对于 LED 的呼吸灯来说，最大值可以取到 255，因此在这里运用一个映射，即使 i 因循环本身最大只能取到 179，但通过一一映射， i 的实际值可以取到 $[0, 255]$ 。

2.1.4 文本

文本的具体内容为代码的文本输出功能，具体包括：文本输入、文本连接、文本转数字、数字转文本、文本长度获取、文本比较。



图 2-1-4-1 文本功能模块

所有文本内容可通过串口监视器或外接显示屏打印。

2.1.5 数组

数组是为了便于代码编写时所采用的一种数字块。具体包括：定义数组、取数组值、改数组值。



图 2-1-5-1 数组功能模块

数组像一个大盒子，可以储存一定个数的数字（第一个指令）或字符串（第二个指令）。在课程中的音乐盒一课，我们就用到了数组。

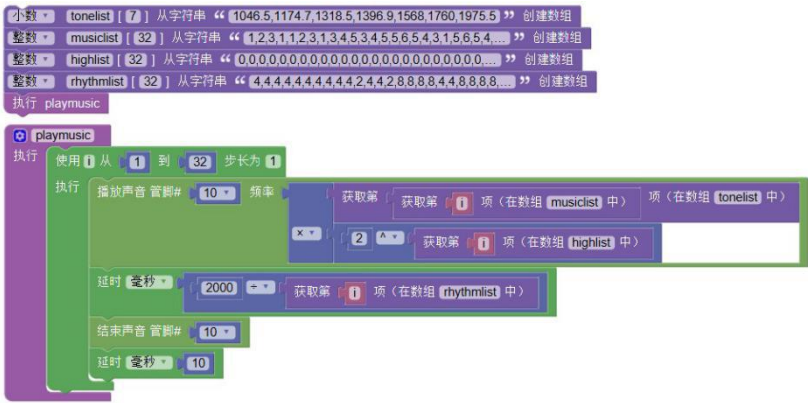


图 2-1-5-2 《两只老虎》代码示例

如上图，tonelist, musiclist, highlist, rhythmlist 分别储存了歌曲《两只老虎》的基本频率（即哆来咪等七音的对应频率），频率变化（即两只老虎的简谱数字），每个音的音名 CDEFGAB 以及每个音的时值（长短）。

《两只老虎》共有三十二个音，因此使用了一个从 1 到 32 的循环语句以此读取第一

个音的频率变化（简谱，并通过



语句可读取简谱的基本频率)，随后与获取 highlist 里的对应音高并按 ththmlylist 里的时值进行相应延时。

该代码将《两只老虎》的指令放到了一个名为 playmusic 的函数内，并通过执行 playmusic 调用该函数。关于函数的内容，我们将在下面函数部分具体详解。

2.1.6 逻辑

逻辑模块中的指令大多是逻辑运算处理内容，具体可分为：条件判断、逻辑运算、？运算。



图 2-1-6-1 逻辑模块

2.1.7 串口

串口模块中的指令都是处理与串口内容相关的问题，分为：设置波特率、串口打印（可加换行）、读数据、串口中断设置、软串口设置



图 2-1-7-1 串口模块

假设学生将 LM35 的温度传感器接到模拟管脚 A0 口，学生可通过

Serial 打印（自动换行） 模拟输入管脚# A0 指令观察当前室温。随后可能会提出疑问：当前室温怎么可能是 58？这将引发学生进一步的思考造成该数据的可能性。

2.1.8 通信

2.1.8.1 通信——红外通信



图 2-1-8-1 红外通信模块

上图指令主要运用的是红外接收管与遥控器之间的数据发射与接收功能。学生掌握了红外通信的相关内容，便可以制作遥控风扇、遥控汽车等自主设计更强的创意电子产品。

2.1.8.2 通信——I2C 通信

I2C (Inter-Integrated Circuit) 是同步通信的一种特殊形式，具有接口线少，控制方式简单，器件封装形式小，通信速率较高等优点。Mixly 支持 I2C 的读取和写入，并且支持基于 I2C 协议的执行器。

2.1.8.3 通信——SPI 通信

SPI 是串行外设接口 (Serial Peripheral Interface) 的缩写。SPI，是一种高速的，全双工，同步的通信总线，并且在芯片的管脚上只占用四根线，节约了芯片的管脚，同时为 PCB 的布局上节省空间，提供方便，拥有简单易用的特性。用户可以使用 Mixly 向 SPI 传输数据。

2.1.8.4 通信——UART 通信

串口通信功能学生可在串口监视器中查看。这一功能对于学生检查自己代码以及监视传

传感器数据提供了便利条件。

软串口：



图 2-1-8-2 UART 通信

2.1.9 存储（高级视图）

存储模块：EEPROM 读写，SD 卡写入



图 2-1-9-1 存储功能模块

2.1.10 传感器

在传感器模块的这个部分中，我们纳入了各种各样的传感器，这些传感器可以分为 2 类：数字传感器、模拟传感器。相信这些传感器一定能给同学们带来许多新奇的体验。

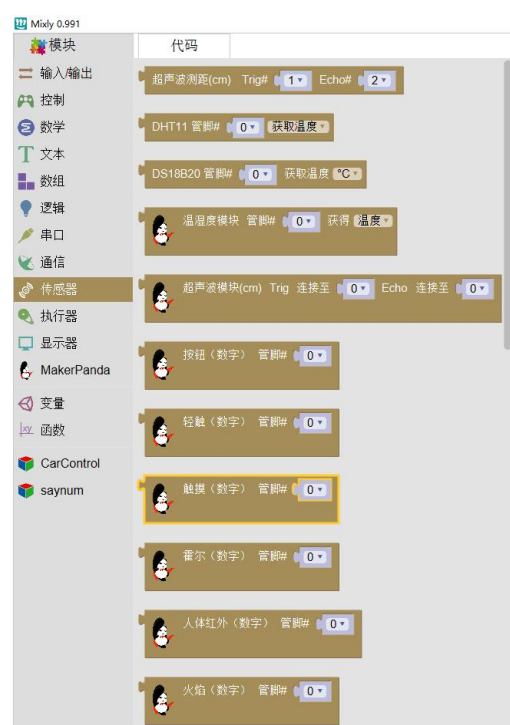


图 2-1-10-1 传感器模块 1

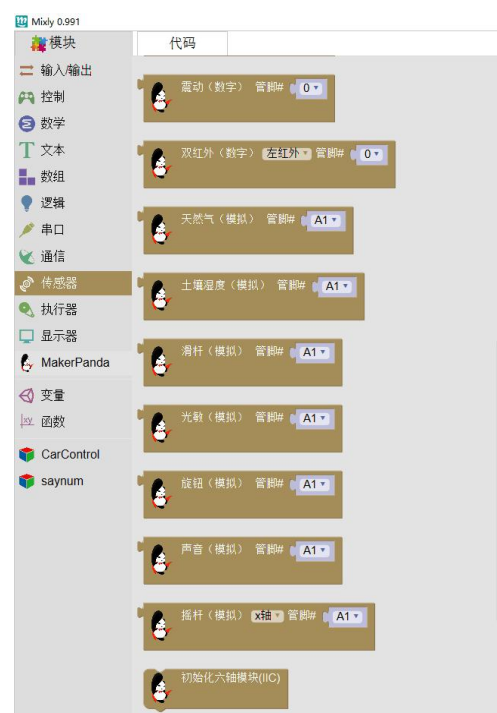


图 2-1-10-2 传感器模块 2

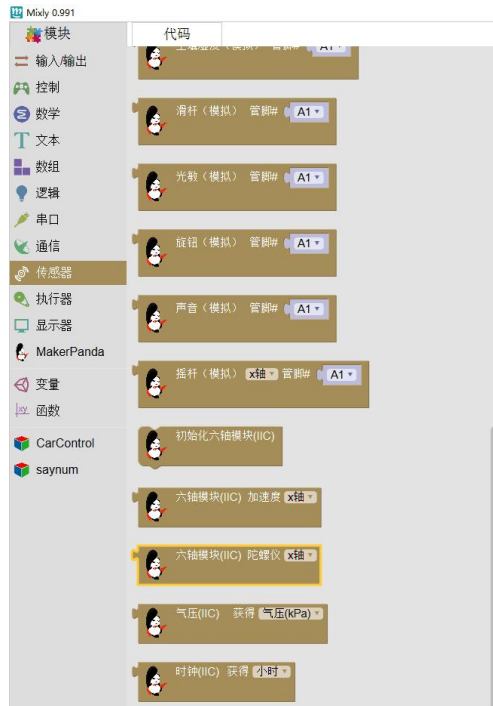


图 2-1-10-3 传感器模块 3

数字传感器包括按钮传感器、轻触传感器、霍尔传感器、人体红外传感器、震动传感器、双红外传感器。数字传感器的返回值都是 0 (LOW) 或者 1 (HIGH)，可以使用 MakerPanda-ONE 主控板上的所有管脚进行连接。例如按钮传感器（编程示例如下图 2-10-1），按一下传感器的按钮就能够在串口监视器看到得到的返回值，串口监视器如下图（2-10-2）。其他的数字传感器类似。



图 2-1-10-4 按钮传感器编程示例

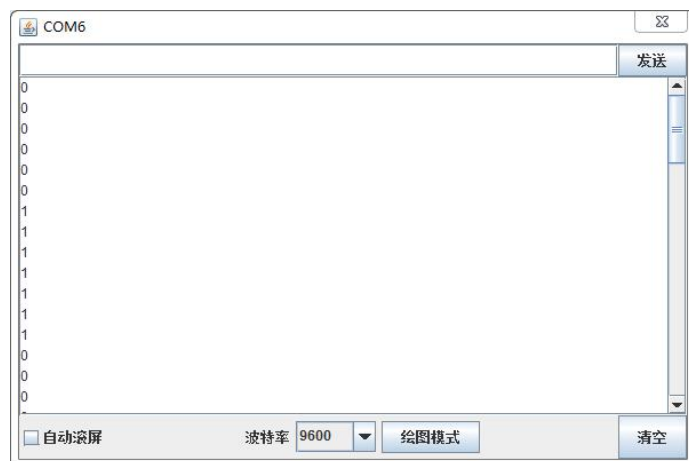


图 2-1-10-5 按钮传感器串口监视器返回值

模拟传感器包括天然气传感器、土壤湿度传感器、滑杆传感器、光敏传感器、旋钮传感器、声音传感器、摇杆传感器、六轴模块（包含多个小模块）、气压传感器、时钟模块。这些模拟传感器返回一个数值（整数或者小数），可以使用 MakerPanda-ONE 主控板上的 PWM 管脚进行连接。例如摇杆传感器（编程示例如下图 2-1-10-3），上下或者左右拨动摇杆传感器会得到 X 轴和 Y 轴的两个方向的值，返回值结果如图 2-1-10-4 所示。其他的模拟传感器类似。



图 2-1-10-6 摇杆传感器代码示例

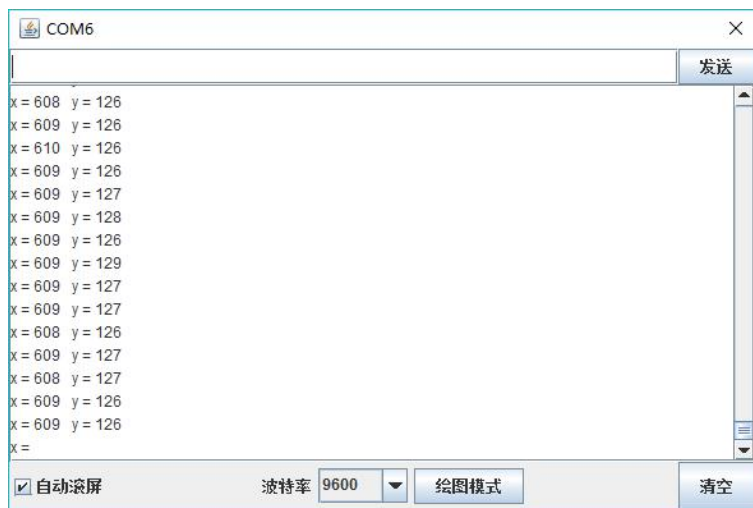


图 2-1-10-7 遥杆传感器串口监视器

2.1.11 执行器

在执行器模块中，我们纳入了许多可以干实事儿的传感器：LED 灯、灯带、9V 继电器、震动马达、风扇、蜂鸣器发声/结束声音、RGB 灯、舵机。



图 2-1-11-1 执行器模块 1



图 2-1-11-2 执行器模块 2

例如 RGB 灯，它通过设定三原色（R、G、B）的数值，使得 RGB 灯发光。它可以使用 MakerPanda-ONE 主控板上的模拟输出管脚进行连接。编程示例如下图 2-1-11-1 所示。



图 2-1-11-3 RGB 代码示例

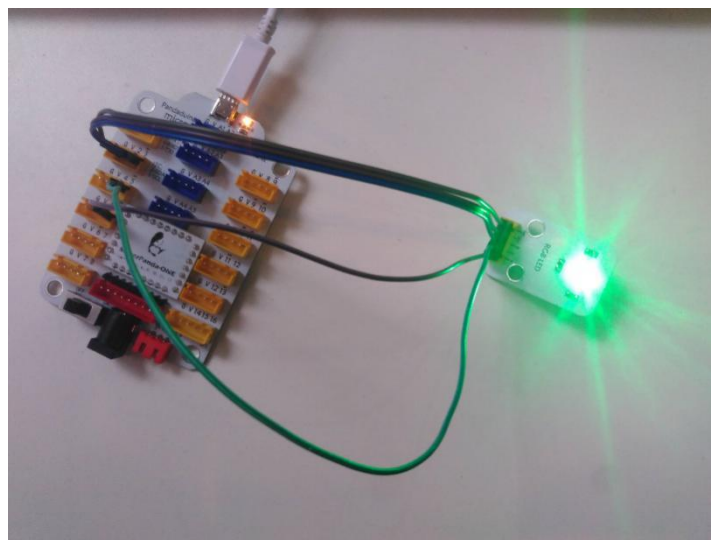


图 2-1-11-2 RGB 实物颜色图

像上面这样通过设置三原色的数值就可以得到你想要的颜色。其他的执行器也有很多功能，你值得一试。

2.1.12 显示器

在显示器模块中，我们纳入了一些可以显示的屏幕：四位数码管、OLED、液晶显示屏。



图 2-1-12-1 显示器模块 1

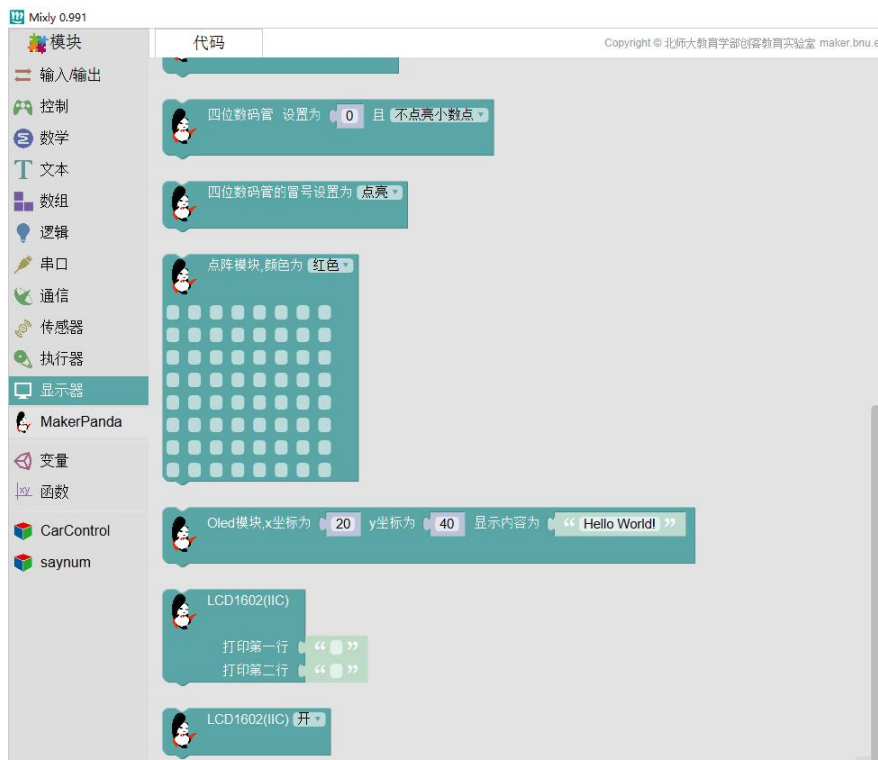


图 2-1-12-2 显示器模块 2

例如液晶显示屏（LCD1602）通过设置显示的文本，我们通过编程实验显示屏的显示。

它必须使用 MakerPanda-ONE 主控板上的 IIC 管脚进行连接。编程示例如下图 2-12-1, 所示。

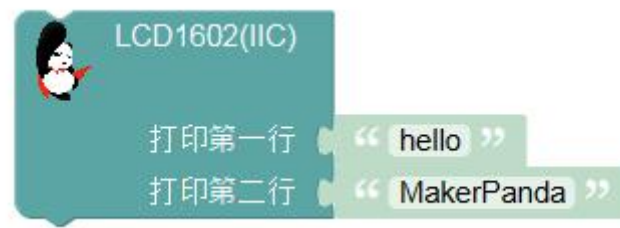


图 2-1-12-1 液晶显示屏的代码示例

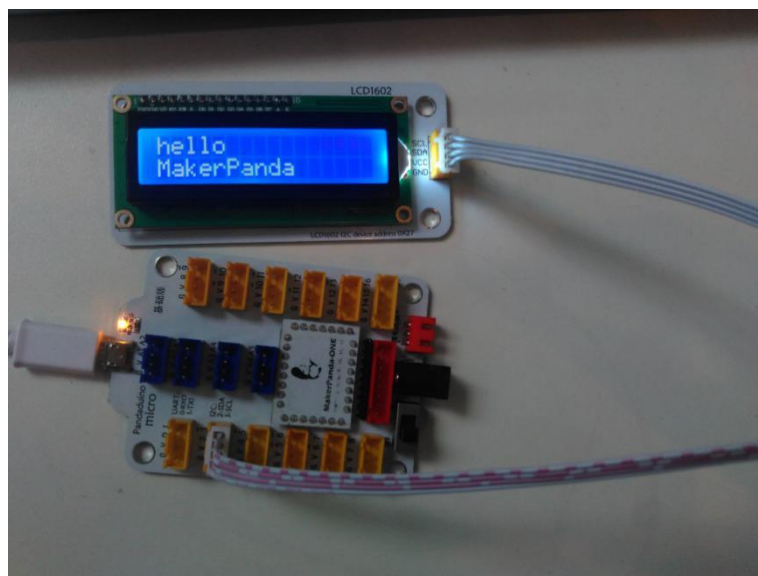


图 2-1-12-2 液晶显示屏（LCD1602）显示图

2.1.13 以太网（高级视图）



图 2-1-13-1 以太网模块

2.1.14 变量

依据变量类型可分为：浮点变量、整型变量、布尔变量、字符串变量。根据变量的值又可分为高或低、真或假。



图 2-1-14-1 变量模块

2.1.15 函数

在函数部分，主要分为定义函数和执行函数，需注意，当用户自定义一个函数时，执行函数的代码是系统生成的。



图 2-1-15-1 函数模块

比如让 LED 点亮这个代码，当我把它放到函数 blink 中后，系统会自动生成执行 blink 代码。



图 2-1-15-2 系统自动生成执行函数 blink 代码模块

另外需要注意，如果使用函数指令，必须运行执行 blink 才可以。如果只自定义函数

blink 而不执行，LED 是不会有反应的。

2.1.16 Factory（高级视图）



图 2-1-16-1 Factory 模块

2.2 程序构建区



图 2-2-1 程序构建区



通常把能完成一定功能的代码在该区域连接。

该区域的右下角有个垃圾桶，可以将不用的代码拖进去删除，也可以将不用的代码直接往最左方拖拽（模块选择区），同样可以删除代码。当然，点击不用的代码后点击键盘 Delete 也可将代码删除，还可以右键点击块选择删除块，见下图 3-1。



图 2-2-2 删除块的方式

在垃圾桶的上方有两个图形，一个是  一个是 。该图形可放大或缩小程序构建区模块的大小。

在  的上方有一个  按钮，这个按钮的功能是将模块的大小恢复至标准大小，并居于程序构建区的中央。点击后模块会大小正常化并出现在程序构建区的中心位置。

在该区域的右上角，还可以选取语言的种类，截至目前为止，米思齐可支持英语、西班牙语、简体中文及繁体中文。在选择语言的左边还有两个箭头，这两个箭头的功能：撤销（undo，Ctrl+Z）及重做（redo，Ctrl+Y）。

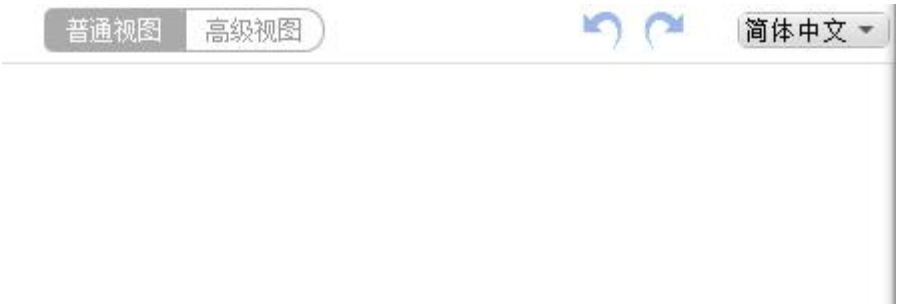


图 2-2-3 程序构建区右上角

撤销功能是在编写代码时误删模块后，便可以点击左箭头或者直接按 Ctrl+Z 来恢复误

删代码；而重做则是和 Ctrl+Z 相反，它是恢复上一步操作。该功能学生可通过点击右箭头或者直接键入 Ctrl+Y 来实现

点击普通视图和高级视图来进行视图之间的切换。



图 2-2-4 程序构建区左上角

点击代码可以对当前模块自动生成的代码进行修改，修改后按照修改的代码进行编译或者上传操作。



图 2-2-5 程序构建区代码修改部分

2.3 代码程序区（可隐藏）

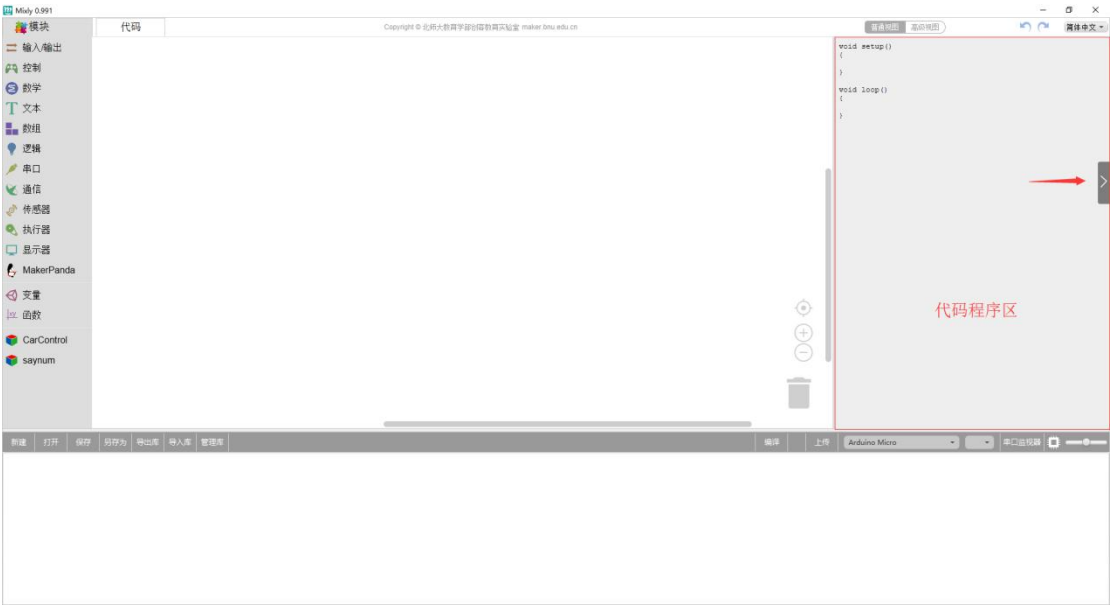


图 2-3-1 代码程序区

该区域可通过点击右侧深灰色箭头来显示或隐藏。在程序建构区拖拽指令后，可以在代码指令区进一步学习程序语言。代码指令区的所有文字代码都是随着相应指令被拖拽而直接生成，即每拖拽一条指令，都会在代码指令区立即反馈。这对于自主学习程序语言、代码自检有着重要作用。

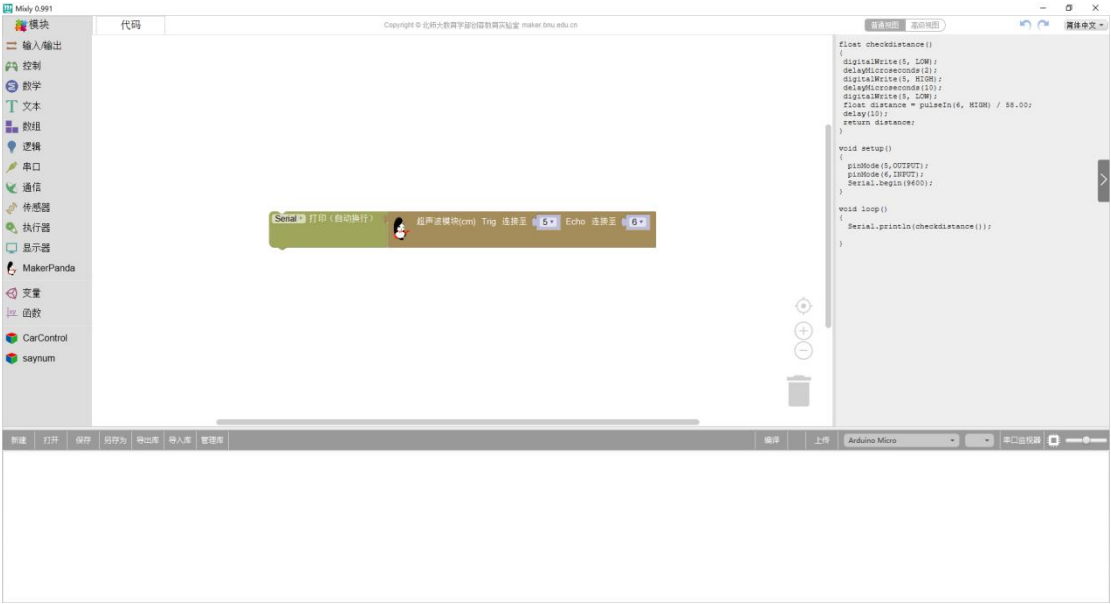


图 2-3-2 拖拽指令即生成相应代码程序

2.4 系统功能区



图 2-4-1 系统功能区

系统功能区主要执行的功能有新建、打开、保存、另存为、导出库、导入库、管理库、编译、上传、选择主控板型号及端口、串口监视器及软件界面放大缩小等功能。

2.4.1 一般功能

新建、打开、保存、另存为是 Mixly 的一般功能，其中代码保存或另存为的格式为.xml。当用户需要打开已保存的文件时，必须通过 Mixly 软件将其打开。也就是说，想要打开某一代码，必须要先打开 Mixly 软件后点击“打开”按钮，找到该代码位置后打开。如果直接点击该代码的.xml 文件，则只能打开该文件的源代码。

2.4.2 库功能

2.4.3 编译&上传

当编写完代码后，如果想要检查代码逻辑是否有误，可点击编译。如果显示“编译失败”，则需要根据提示检查自己的代码，如显示“编译成功”则证明代码逻辑上无误，可上传。如果出现“上传失败”，大多数情况插拔 USB 线即可解决该问题；如果出现“上传成功”，则证明代码已上传至板子上。当然，如果对于代码逻辑信心十足，可直接点击上传按钮。

2.4.4 主控板选择&端口

当点击主控板下拉三角时即可看到有众多主控板型号可供选择。需按照当前手中主控板型号予以选择。MakerPanda 套件使用的主控板型号为 Aduino Micro。

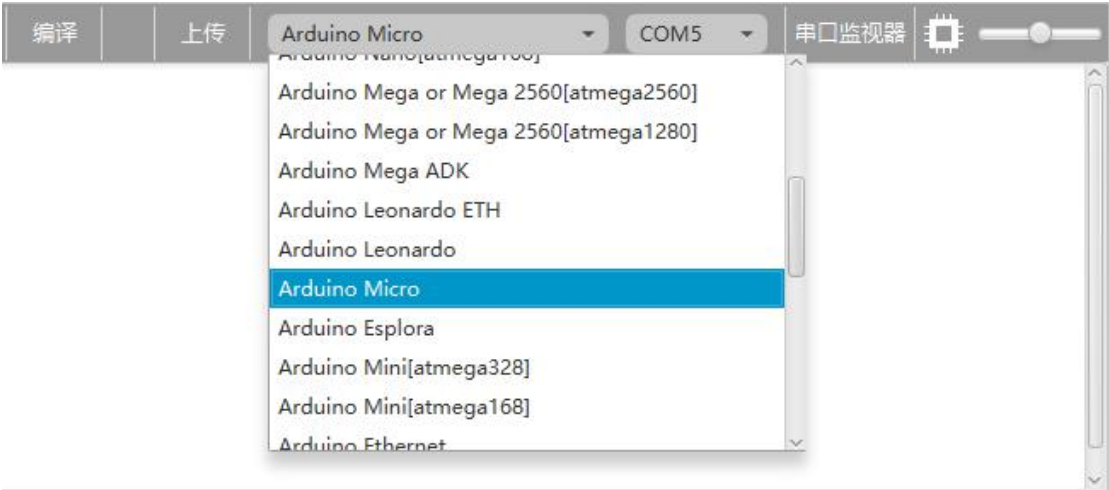


图 2-4-4-1 主控板选择

上传前，需要将主控板与电脑通过串口线连接，此时将显示具体的端口号。

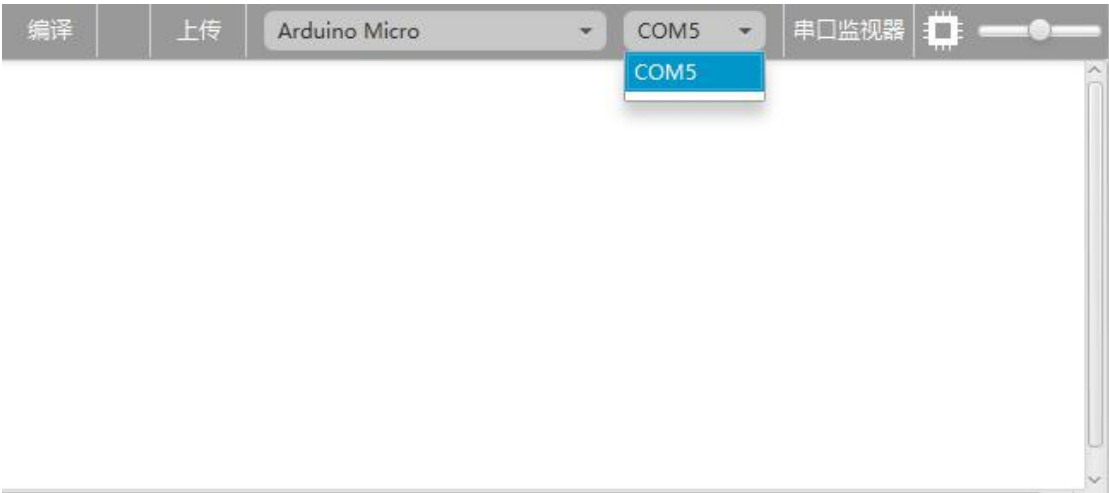


图 2-4-4-2 端口选择

2.4.5 串口监视器

串口监视器与模块选择区的通信模块中的串口通信指令一起使用。假设学生已连接红外接收管，他可通过串口监视器时时观测遥控器上不同按钮的十六进制数（其中，只有 FD 开头的数才是真正的接收数值，其他数据均为干扰数据）。

需要注意的是，在使用串口监视器后需及时关闭。如果在没有关闭串口监视器的情况下进行代码的再次上传，往往是失败的。因为串口监视器工作时占用了主控板的相关端口，使得上

传失败。

2.4.6 拖动缩放界面按钮

在串口监视器右侧有一个左右可拉的按钮，该按钮可放大或缩小 Mixly 整个界面的大小。

与程序构建区内的 $\bigcirc+\bigcirc-$ 不同（该按钮只可放大或缩小程序构建区代码的大小），拖动缩放界面可放大除消息提示区外所有区域的大小。

2.5 消息提示区

消息提示区通常是给予以信息反馈的场所。比如编译或上传进程中，编译或上传是否成功，如果失败原因是什么；或者是导入库是否成功等消息。



图 2-5-1 消息提示区