|  |  |  |
| --- | --- | --- |
| 课程主题 | | 屏幕输出 |
| 任务描述 | 1.通过Python编程，让机器人屏幕输出一行字“hello，world” | |
| 知识点 | 马达控制方式，屏幕控制，播放声音 | |
| 场地 | 不需要 | |
| 搭建 | 不需要 | |
| 程序 | #!/usr/bin/env pybricks-micropython  from pybricks import ev3brick as brick  from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,  InfraredSensor, UltrasonicSensor, GyroSensor)  from pybricks.parameters import (Port, Stop, Direction, Button, Color,  SoundFile, ImageFile, Align)  from pybricks.tools import print, wait, StopWatch  from pybricks.robotics import DriveBase  # Write your program here  brick.sound.beep()  brick.display.text("hello,world",(10,10))  wait(5000) | |
| 教学过程 | | |
| 说明：把三个问题放到一节课中来实现，内容有些多。我们一点儿一点儿的去设计，看看如果需要分开的话，我们考虑把内容分开。  1.新建一个工程（create a new project）。  在VS Code中，如果要给EV3写程序，一定要先建立一个工程。这一点儿跟EV3的图形化编程是一致的，只不过图形化软件中，这一步是自动完成的，而在python中，需要我们手动完成。  第一步：打开VS Code，点击左侧的EV3标签图标。切换到下面的界面。    第二步：点击Create a new project，然后弹出一个小提示窗，输入工程文件名：  这里我们输入MotorTest，（中文意思是马达测试，请记住：一定是英文名称，EV3主机不认识中文）。  完成后敲击键盘上的回车键（Enter）  这个时候会弹出一个窗口，如下所示：    让你选择一个文件夹，这个文件夹是用来存放工程文件的，文件夹的内容，直接下载到EV3主机中去。  我们一般选择新建文件夹，然后选中这个文件夹，点击窗口右下角的“Select Folder”。比如这里，我们新建一个MotorTest文件夹。    完成之后，点击窗口右下角的“Select Folder”，VS Code 侧边栏就产生了变化，现在列出的是我们新建的工程文件的所有内容，很多都是自动为我们建立的。    其中，main.py是我们用来控制EV3主机的文件，很长一段时间，我们所有的程序代码，都是写在这里面的。我们打开看看。  二、分析main.py  打开main.py,我们看到如下界面。    就像我们打开EV3编程软件，会看到有一个程序开始的模块在程序中一样，main.py中，已经为我们写好了很多代码。这些代码提供了最基本的程序运行所需要的Python库文件，我们现阶段可以不必纠结上面的这些代码。随着学习的深入，我们再回过头来研究这些内容。  简单说一下这些语句的作用吧：  第1行，是一个注释，说明了这个文件的位置和用途  第3-9行，是引入EV3运行所需要的基本库文件  第11行：注释，告诉我们这一行的下面可以写我们自己的代码了。  第12行：代码行，让EV3主机发出一个声音。  从第13行开始，就可以人有我们驰骋了，但就像骑马要懂基本的规则一样，我们的代码也需要按照一定的要求来写。  第三步：打个招呼吧——屏幕上输出：Hello，World。  从C语言开始，学习编程基本都是从Hello，World开始（可以百度一下hello，world，了解一下历史趣事），我们也不例外。  尝试着在main.py的第13行和第14行输入下面的语句：    让我们尝试着运行一下吧：    点击左侧的小虫子图标，切换到调试界面，然后点击左上角的绿色是三角符号。  出现了一个问题，右下角告诉我们没有连接EV3主机，需要我们先完成电脑和EV3的连接。    我们点击一下：Connect Now，看看会有什么情况出现。    顶端的变化，让我们选择一下，当然选择是：ev3dev这个选项了，点一下看看。  连接好了，貌似机器人也发出了响声，但没有显示任何内容。仔细检查一下窗口的变化，我们发现，输出窗口变得比较大，并且好像告诉我们哪里出错了。  这里的关键信息，line 13，是告诉我们13行，有问题。最后一行，Exited with error code 1.  说的是因为的一个错误，程序退出运行。  修改程序吧，查阅资料后，进行如下修改：    对照上面的代码，很容易发现，这里第十三行，我们把10，10这两个数字用括号括起来了。  再运行。  如果一直盯着EV3主机的屏幕，你会发现，屏幕上有信息出现了，然后一闪就过去了。  再看看程序窗口的输出：    我们看到了Complete successfully. 的字样，这说明程序运行成功。  四、对程序的修改：  运行以上程序，我们至少这么些疑问：  1.EV3屏幕显示的内容，很快就过去了，能不能让显示的时间变得长一些？  2.hello， world显示的位置太靠上，能不能显示在屏幕中间？  3.上面代码中，为什么不加括号就错了，加了括号就好了呢？  我们尝试着一个个的把问题解决：  **第一个问题**，其实是显示时间过短造成的，我们修改第14行的wait()中的数值，就可以解决，比如将1000改为5000，再运行程序看看。  **第二个问题**：其实和第三个问题是联系在一起，在语句brick.display.text("hello,world",(10,10))  中，两个数字，是用来控制hello，world的显示位置的。我们尝试着改一改，然后总结一下，这两个数字分别代表的是什么。  修改（10，10）为以下的几组数据，挨个尝试：（100，10），（100，50），（100，100）  （10，60），（50，60），（100，60）……  找到你想要的规律了吗？  EV3屏幕是由一个一个的点组成的，从左到右，每行178个点，从上到下，每列128个点。  那么，要想让hello，world显示在中间，我们该怎么做呢？尝试一下吧。  第三个问题：用括号把这两个数字括起来，这是一个语法的要求，这里的用途是提供两个数据，分别表示行和列的起实位置。  五：练习：   1. 在屏幕上输出“I am EV3“ 2. 在屏幕上输出两行内容，第一行是：EV3，I’m coming ;第二行是：Python, I’m coming 3. 圣诞节快到了，在在屏幕上输出一棵用“\*“表示的圣诞树吧   六：小结：  这里有几个单词是要记住的：  brick：乐高机器人的主机的昵称，砖块  display：显示，在这里，指的是python中的一个显示类  text：文本，是display的一个方法，在屏幕上把指定的内容显示出来。  wait：一个函数，几乎所有的代码编程语言中，都用这个表示让机器人持续一个动作的时间。 | | |
| 注意事项 |  | |
| 课程修改备注 | 课程贡献者： | |
| 课程整理者： | |
| 课程修订者：   1. XXX在20XX年XX月XX日，对XX进行了修订，修改了XX内容，增加了XX内容 2. XXX在20XX年XX月XX日，对XX进行了修订，修改了XX内容，增加了XX内容 | |