

高中 信息技术教材

实践项目指南

教师
用书



掌控板

感知环境促健康

一、项目前置知识简析

本项目配合教材必修一课程使用；
基于核心知识点：

数据与信息

教学需要的前置知识点有：

1. 数据特征
2. 信息特征

项目实践内容主要包括：

1. 学习如何使用传感器采集数据；
2. 学习从数据中获取信息；
3. 理解数字量和模拟量的概念；
4. 理解数据与信息的概念和他们之间的关系；
5. 学习计算思维的概念，理解如何借助计算思维解决计算问题。

二、教学目标

1. 根据连线图进行实物连接，理解传感器的使用和接线方法；
2. 理解数据采集的方法；
3. 掌握在 BXY 中编写代码及查看数据的方法；
4. 掌握从数据中获取信息的能力。

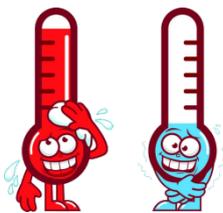
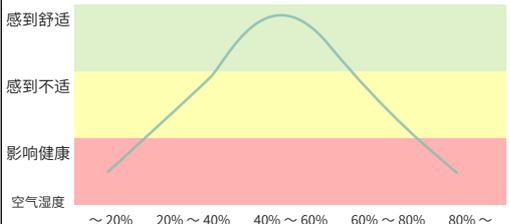
三、教学重难点

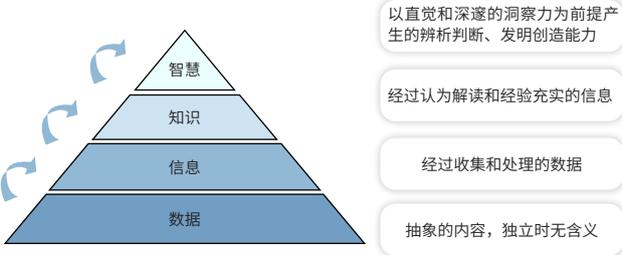
教学重点：根据连线图进行实物连接，理解传感器的使用方法。

教学难点：掌握在 BXY 中编写代码读取传感器数据。

四、教学流程

1. 背景分析，引入目标

教学环节	教学内容	学生活动	设计意图
情景导入 1	 <p>背景故事阅读：</p> <p>大家都知道，在气温过高和过低的时候都会影响到我们身体的状态，甚至可能会生病。但是你知道吗？其实除了温度，还有其他因素会影响到我们的身心状态。</p> <p>在炎热的夏季，很多人都会觉得闷热难忍，身体乏力，而在冬季，很多人又会觉得干燥上火，大多数人认为这是“温度”在作祟。其实，影响我们身体健康及身心健康的并不仅仅是温度，空气湿度也是造成这一现象的主要原因。</p> <p>提出问题并讨论：</p> <ol style="list-style-type: none"> 大家知道什么空气湿度是随着什么变化的吗？（会根据一年四季的气候变化而变化。） 那么湿度的多少会影响到我们的身体健康呢？（引入到下阶段资料阅读） 	参与讨论，并记录讨论结果，进行分享。	通过讨论影响身体状况的因素，引出湿度的条件。
情景导入 2	 <p>湿度对人体健康的影响资料阅读：</p> <p>人体感觉比较舒适的湿度是 40%-60%，在空气湿度为 55% 时，病菌较难传播；但如果空气湿度超过 65%，人会感到胸闷、呼吸困难，湿度过高时，潮湿的空气容易让人患风湿病和气管炎，相对湿度如果达到 80% 以上，有碍人的机体蒸发散热，对患有肾病、结核病、慢性腿腿病的人都有不良影响，如超过 90%，会使人体呼吸系统和黏膜产生不适。</p> <p>当室内空气湿度低于 40% 的时候，灰尘、细菌等容易附着在黏膜上，刺激喉部，引发咳嗽，同时容易诱发支气管炎、哮喘等呼吸系统疾病。湿度过小时，因上呼吸道粘膜的水分大量丧失，人感觉口干舌燥，甚至出现咽喉肿痛、声音嘶哑和鼻出血，并诱发感冒。</p>	参与讨论，并记录讨论结果，进行分享。	在了解了空气中的温湿度条件对健康的影响，引出本节课的目标，设计环境检测装置的任务。

	<p>提出问题并讨论：</p> <p>1、那么，在了解了湿度对人体的影响后，猜猜现在的温度与湿度大概是多少？记录在手册上。</p> <p>2、结合温度与湿度，分析一下什么样的温湿度是会让人感觉舒适的，什么样是会影响健康的。</p> <p>提示：</p> <p>统计记录大家的猜测的温湿度与适宜的温湿度，可以与后续实际测得的数据进行对比。</p>		
引出目标	<p>教师总结：</p> <p>在这里，温度与湿度的数值就是“数据”，而我们通过这两个数值分析得出对健康有影响的状态就是“信息”，当我们积累足够多的“信息”，这些信息也就是我们自己的“知识”了。</p>  <p>数据、信息、知识与智慧的递进关系</p> <p>通过以上的学习，大家了解到温湿度对健康的重要性。我们需要一个能够检测环境温湿度并且能够在环境变差的时候提醒我们的装置。</p>	回顾数据、信息、知识与智慧之间的关系并思考本节课的任务如何设计。	明确目标为下一阶段任务做准备。

2. 项目实践

对学生进行分组，分为三人一组完成项目。讨论对该装置的**功能需求**。

教学背景资料 - 计算思维：

在课程种我们将以计算思维作为主要分析工具使用，下面介绍一下什么是计算思维。

2006年3月，美国卡内基·梅隆大学计算机科学系主任周以真（Jeannette M. Wing）教授在美国计算机权威期刊《Communications of the ACM》杂志上给出，并定义的计算思维（Computational Thinking）。周教授认为：计算思维是运用计算机科学的基础概念进行问题求解、系统设计、以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

计算思维是运用计算机科学的基础概念进行问题求解、系统设计、以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。2010年，周以真教授又指出计算思维是与形式化问题及其解决方案相关的思维过程，其解决问题的表示形式应该能有效地被信息处理代理执行。

计算思维是一种解决问题的思维过程，可划分为四个步骤：

- (1) 分解：把数据、过程或问题分解成更小的、易于管理的部分。
- (2) 模式识别：观察数据的模式、趋势和规律。

- (3) 抽象：识别模式形成背后的一般原理。
 (4) 算法设计：为解决某一类问题撰写一系列详细步骤。

对这四个步骤的理解过程是：

- (1) 分解，是大化小，将一个复杂问题分解为简单问题。
 (2) 模式识别，是小并小，分析理解简单问题的实质，寻找问题之间的联系。
 (3) 抽象，是概括小，高度概括简单问题的实质，为高效解决问题指引方向。
 (4) 算法设计，是解决小，用切实可行的方法，解决小问题，以达到解决复杂问题的目标。

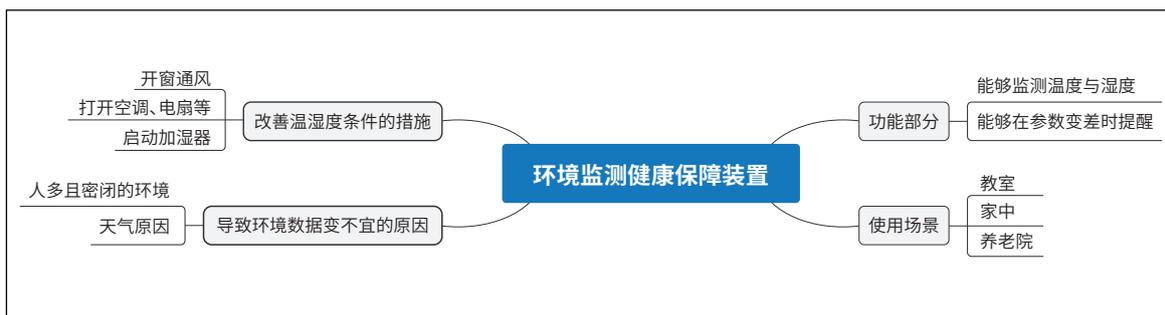
计算思维是现在学校教育提倡培养学生的核心要素。高中信息技术的核心要素包括：信息意识、计算思维、数字化学习与创新和信息社会责任。

注意是计算思维而不是计算机思维，虽然计算思维在设计程序的时候非常有用，但是它并不是只为计算机为生的思维方式，它是诞生于信息化社会的思维方式。利用计算思维，可以帮助我们更好的适应信息化时代，更好的解决信息化时代所面临的各种问题。帮助学生认识并学习这一思维方式符合信息技术课程的学科特点。

本项目作为课程中的第一课，没有涉及过多算法知识，所以不会用到完整的计算思维的内容和流程，完整的计算思维内容会在后续课程中逐渐丰富。本项目主题为探究为主，所以会使用分析探究类的思维工具辅助。

环节 1 问题分解

教学环节	教学内容	学生活动	设计意图
环节 1 问题分解	<p>根据计算思维解决问题的流程，第一步就是问题的分解。</p> <p>为了方便分析，带领学生学习使用思维导图工具。</p> <p>思维导图工具介绍： 思维导图，英文是 The Mind Map，又叫心智导图，是表达发散性思维的有效图形思维工具，它简单却又很有效，是一种实用性的思维工具。</p> <p>思维导图运用图文并重的技巧，把各级主题的关系用相互隶属与相关的层级图表现出来，把主题关键词与图像、颜色等建立记忆链接。思维导图充分运用左右脑的机能，利用记忆、阅读、思维的规律，协助人们在科学与艺术、逻辑与想象之间平衡发展，从而开启人类大脑的无限潜能。思维导图因此具有人类思维的强大功能。</p> <p>思维导图是一种将思维形象化的方法。我们知道放射性思考是人类大脑的自然思考方式，每一种进入大脑的资料，不论是感觉、记忆或是想法——包括文字、数字、符码、香气、食物、线条、颜色、意象、节奏、音符等，都可以成为一个思考中心，并由此中心向外发散出成千上万的关节点，每一个关节点代表与中心主题的一个连结，而每一个连结又可以成为另一个中心主题，再向外发散出成千上万的关节点，呈现出放射性立体结构，而这些关节的连接可以视为您的记忆，就如同大脑中的神经元一样互相连接，也就是您的个人数据库。</p> <p>教师根据以下示例，引导学生学习思维导图的使用及应用思维导图时的思路。</p>	<p>学生利用思维导图工具分析装置的功能需求和设计需求，在学生手册中完成并分享。</p>	<p>在此环节学习思维导图工具来训练学生的问题分解与关联的能力。并完成计算思维第一步的问题分解任务。</p>



思维导图工具介绍:

应用思维导图时，会根据主题进行发散思考，也就是对主题问题的拆分，一般在分析这类问题时，我们首先会考虑这个装置的功能需求，需要有哪些功能，这就是对该问题的第一次拆分；然后就要考虑这个装置是用在哪里的，因为不同场景下的使用会有很大的差别；最后我们再根据这个问题的特点，我们这节课需要制作的是检测空气中温湿度的装置，所以会针对这个条件会有一些问题的拆分，如会引起温湿度变化的情况和当条件变差时如何改善的；在示例中给出四个方向的基础上，鼓励学生能够向更多方向发散。

环节 2 模式识别

教学环节	教学内容	学生活动	设计意图
环节 2 模式识别	<p>计算思维的第二步是模式识别，利用 KANO 模型可以快速的实现功能的模式识别。</p> <p>KANO 模型工具介绍：</p> <p>KANO 模型适用于对大量需求进行分类和整理的一种分析模型，能够快速将需求归类整理出产品的核心需求，并筛去非必要甚至有影响产品满意度的需求。</p> <p>根据不同类型的质量特性与顾客满意度之间的关系，狩野教授将产品服务的质量特性分为五类：</p> <p>基本（必备）型需求——Must-beQuality/ Basic Quality</p> <p>期望（意愿）型需求——One-dimensional Quality/ Performance Quality</p> <p>兴奋（魅力）型需求—Attractive Quality/ Excitement Quality</p> <p>无差异型需求——Indifferent Quality/Neutral Quality</p> <p>反向（逆向）型需求——Reverse Quality，亦可以将'Quality'翻译成“质量”或“品质”。</p> <p>前三种需求根据绩效指标分类就是基本因素、绩效因素和激励因素。</p> <p>提示：</p> <p>这里我们将简化 KANO 模型，将需求简单分为：1、基础需求，就是最基础的必要功能；2、期望需求，就是在完成了基础需求的基础上，有这些功能会提升体验感；3、无差异需求，就是无论有没有，并不会影响使用体验的好坏。</p>	按照要求引导完成分析，并在学生手册填写完成表格内容。	进行计算思维第二步的分析，并学习 KANO 模型工具解决实际问题。

KANO 模型表格示例：

项目需求分级表

基础需求	期望需求	无差异需求
测得温度数据 测得湿度数据	可以根据温度变化提醒用户	酷炫的外观



提示：

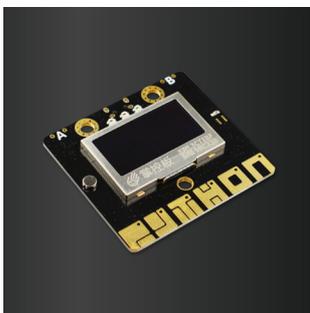
根据分析得出，基础需求为：

- 1、能够监测和显示环境温度与湿度；
- 2、能够提醒人们温度湿度不适宜了。

环节 3 知识学习

教学环节	教学内容	学生活动	设计意图
硬件知识 导入	教师提问： 1、在你们的家里有没有可以检测环境信息的物品？（如温度计、湿度计等） 2、你们知道他们的工作原理吗？ （针对几种常见物品进行简析。水银温度计：根据热胀冷缩的原理来反馈温度；传感器温湿度计：通过传感器检测温湿度进行反馈。） 引出传感器和掌控板编程控制。	讨论问题并记录在学生手册中。	引出传感器与掌控板编程控制。

硬件介绍：



掌控板



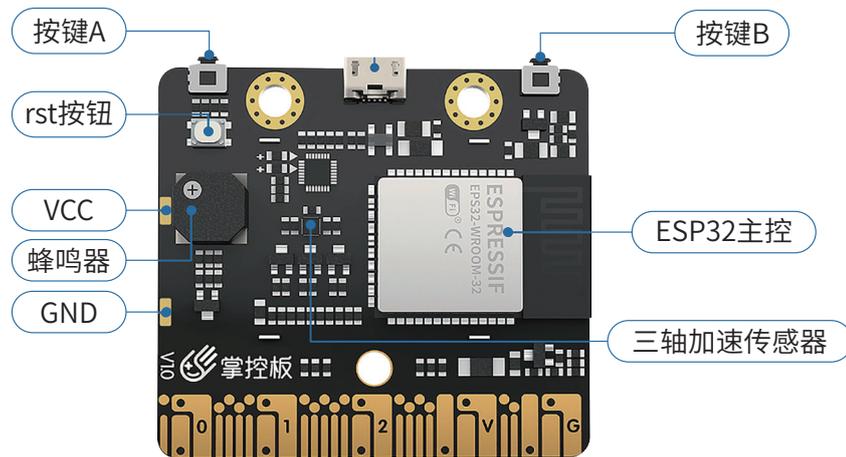
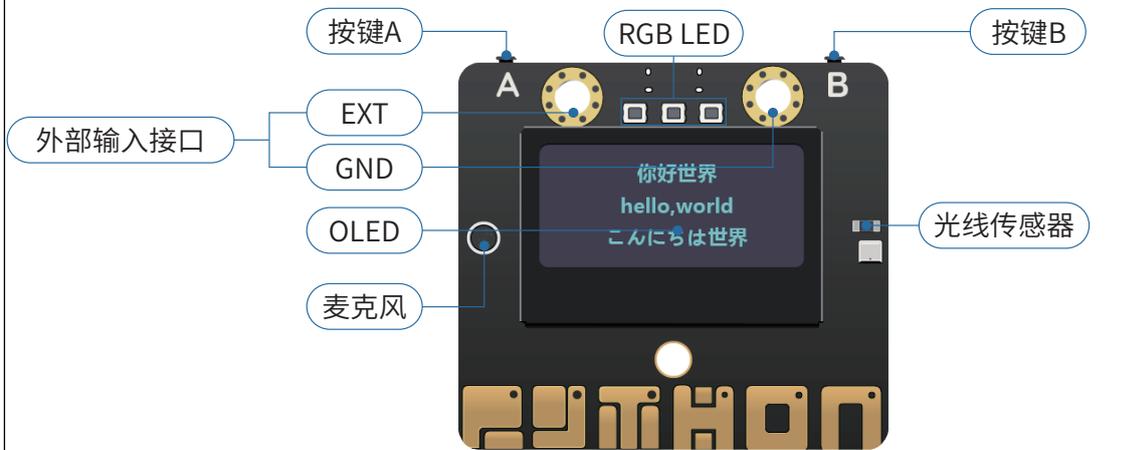
掌控板扩展板



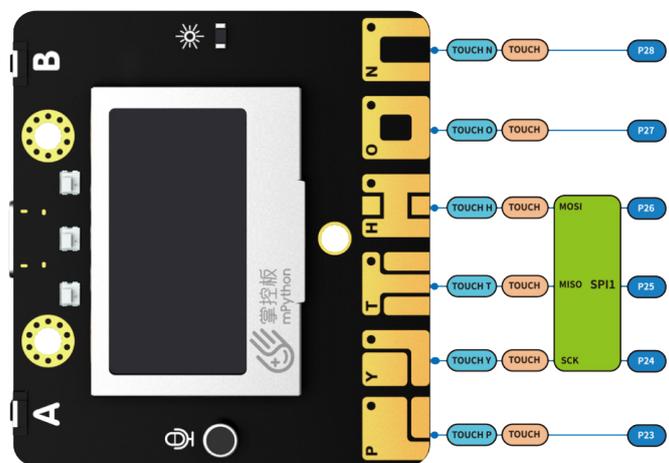
温湿度传感器

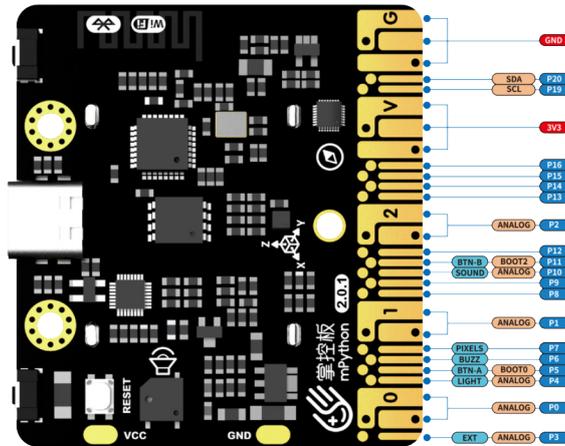
掌控板在这个案例中的作用是作为核心控制板，用于实现我们编写程序中的功能，并且板载模块多，掌控板功能模块如图。掌控板支持 MicroPython 编程，可以通过 Python 语法来编程控制硬件。

掌控板介绍：



掌控板还有很多扩展引脚，功能如图。



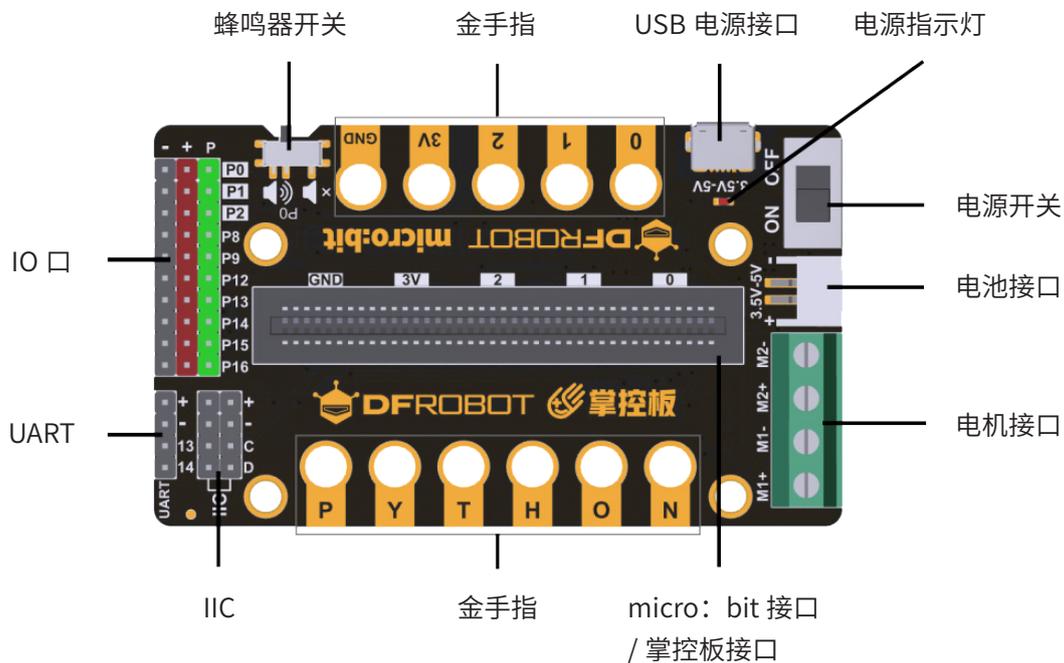


引导环节:

可以看到掌控版上有很多扩展引脚，但是我们要怎么接出来呢？我们需要借助 IO 扩展板来将划分得很细得这些引脚引出来使用。

IO 扩展板介绍:

扩展板的作用是将掌控板上的 IO 口以接线口的形式引出，方便我们接入传感器、执行器等模块。IO 口与掌控板上的接口一一对应。



温湿度传感器介绍:

温湿度传感器是传感器得一种，它的功能则是用于采集温度和湿度数据，让我们可以通过掌控板看到环境得温湿度数据及变化。

连接模块，并进行简单控制读取环境温湿度数据。详细连接与操作参考教程文档。

硬件搭建

硬件清单：

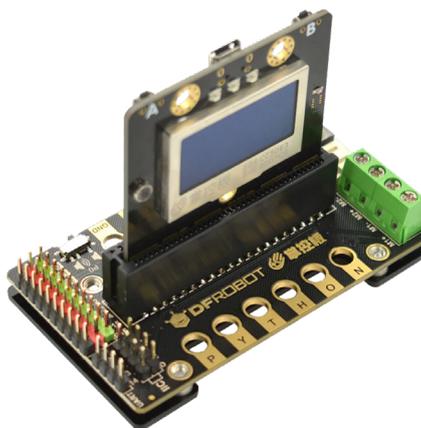
主控：掌控板、扩展板；

模块：温湿度传感器。

通过图示或实物展示：掌控板与扩展板的连接方式（主要是防止主控板插反）和传感器与扩展板的连接方法（防止反接，可能会损伤模块）。

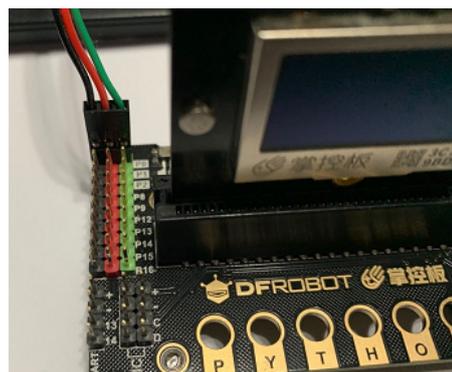
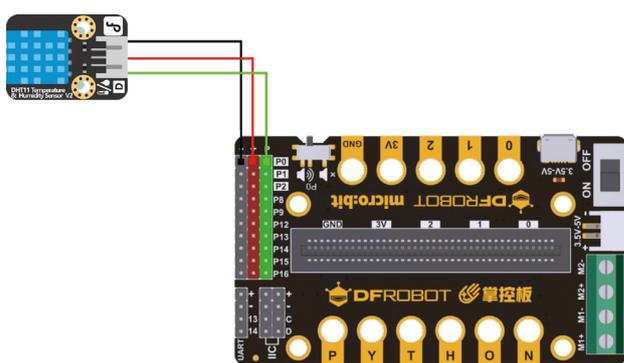
掌控板与扩展板正确结合方式，掌控板的 OLED 屏幕的那一面，对着掌控板的图标那个方向插入。

如图所示：



学生操作：

按照接线图将模块利用连接线组装好。（提示学生注意接线的颜色与接口的颜色对应）



注：此处传感器接到了扩展板的 P0 接口，这里接口与后续编程中程序使用的接口号需要一致。如果这里接到了 P1 在程序中，所有 P0 的指令需要替换为 P1。

程序部分：

本项目使用到的编程软件为 BXY Python Editor（后直接用 BXY 表示），该软件是一款运行于 Windows 平台的 MicroPython 编程 IDE，界面简洁，操作便利。内置了很多基础操作库。为众多 MicroPython 爱好者提供了一个简洁实用的平台。

我们需要下载 MicroPython 编程 IDE-BXY，说明及下载地址：

<http://docs.dfrobot.com.cn/bxy/>



注：BXY 在下载安装时有绿色版和安装版两种可以选择，建议使用安装版，绿色版使用时需要注意，初次开启是没有示例程序的，需要在有网的前提下，首次启动软件一次后重启才可以看到。

软件界面功能如图
详细操作步骤见附录。



程序运行效果：



学习 BXY 软件操作，调试程序，并能通过模块获取温湿度数据。

程序解析：

```

1  from mpython import *
2  from dht import DHT11
3  import time# 导入程序必要的包和块
4  dht=DHT11(Pin(Pin.P0)) # 设置 DHT11 传感器引脚
5  while True:
6      dht.measure()
7      oled.fill(0)
8      oled.DispChar(" 温度 :",0,10)
    
```

```

9      oled.text("%d" % (dht.temperature()), 48, 14)
10     oled.DispChar(" 湿度 :",0,35)
11     oled.text("%d" % (dht.humidity()), 48, 40)
12     oled.show()
13     time.sleep(1)
    
```

教师提问：

传感器获取到的温度和湿度属于什么信号？（数字信号 / 模拟信号）

环节 4 抽象化

教学环节	教学内容	学生活动	设计意图
环节 4 抽象化概念	<p>教师讲解： 在上一步中，我们分析得出了几个具象化的功能，就是每个功能都很具体。可能由于项目内容不是很复杂，所以条件较少，还可以分辨清楚，但是当具象化的条件变多之后，这些条件之间的逻辑关系就会变得复杂，就需要我们将这些问题进行抽象化处理。</p> <p>抽象化概念： 抽象化是提取概念的本质的过程，这样的话就去除了与原来有关联的现实中的对象的依赖关系，并对其进行泛化，使其具有更广泛的应用，从而与其他等效现象的抽象描述相匹配。简单来说就是识别模式形成背后的一般原理。这么看也许很不好理解，下面用示例来增强大家对抽象化这一步骤的理解，如图是将丑小鸭的故事和灰姑娘的故事中的关键点进行抽象化提取，然后进行对比。</p> <p>抽象化案例教学： 问：这里好黑啊。 抽象化结果：需要光源（开灯或手电） 问：我好冷啊。 抽象化结果：需要开空调或加衣服</p>	参与抽象化学习，并对问题进行抽象化处理，记录在学生手册上。	由于本案例对抽象化应用比较简单，这里主要需要学习抽象化的分析方式及使用它的意义。

示例：

我们项目中提取出来的核心需求：

- 1、能够监测和显示环境温度与湿度；
- 2、能够提醒人们温度湿度不适宜了。

经过抽象化之后就是：

- 1、读取传感器数据；
- 2、传感器数值在超过阈值时报警。

序号	条件	抽象化结果
1	能够监测和显示环境温度与湿度	读取传感器数据

环节 5 算法设计

教学环节	教学内容	学生活动	设计意图
环节 5 算法设计	提示： 本环节将任务分为三个阶段，进行算法设计与程序编写，由于现阶段学生还未学习算法相关知识，所以这里的算法设计内容简化，通过问题引导学生讨论，通过实验并记录在手册中。		
算法设计 阶段一 数据测试	教师提问： 如果当温度和湿度到达不宜健康的时候，我们通过设计的装置提醒，然后打开窗户（或其他改变环境条件的操作）这时，教室内的温度和湿度会立刻恢复到合适的数值范围内吗？不会？为什么？ 回答示例： 因为环境数据的变化是一个缓慢变化的过程，不会出现断崖式上升或下降的情况。 教师提示： 如果等到温湿度已经到达“红线”时再提醒，就会有些迟了，所以，我们需要在数据趋近于临界值之前就进行提醒。那么需要提前多久？在什么时候提醒？就需要我们进行对数据的搜集并预估。如： 开窗 10 分钟的温湿度变化前后对比 室内封闭 10 分钟温湿度变化前后对比 打开加湿器 10 分钟温湿度前后对比	每个小组选择不同的数据进行测试，并将内容填写到学生手册中之后将大家得到的结果进行分享，凭借这些数据推算需要在何时提醒“我”环境即将变得“不宜”。	在运行装置时可能会遇到各种各样的问题，所以设计算法之前需要将现实应用中可能出现的问题进行思考，减少问题的发生率。

算法设计 阶段二 设置提醒	教师提问： 掌控板上有这么多执行器可以对我们进行提醒，你所选择得方式是什么？ 提示： 根据小组选择得执行器不同，去查阅资料获取该执行器的使用方法。（掌控板执行器使用案例在学生手册附录）并参考示例程序中 if 函数的使用方法来编写到自己的程序中。	小组讨论决定使用什么功能作为提醒。	此步作为算法设计初步接触了解判断功能为之后学习算法打下基础。
项目示例程序： 示例程序是使用掌控板上方的三颗彩灯实现提醒功能。			
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	<pre> from mpython import * from dht import DHT11 import time dht=DHT11(Pin(Pin.P0)) while True: x = dht.temperature() # 定义温度值为 X y = dht.humidity() # 定义湿度值为 Y dht.measure() oled.fill(0) oled.DispChar(" 温度 :",0,10) oled.text("%d" % (dht.temperature()), 48, 14) oled.DispChar(" 湿度 :",0,35) oled.text("%d" % (dht.humidity()), 48, 40) oled.show() time.sleep(1) if x > 16: rgb[0] = (16 ,16 ,16) rgb[1] = (16 ,16 ,16) rgb[2] = (16 ,16 ,16) rgb.write() if x <= 16: rgb[0] = (0 ,0 ,0) rgb[1] = (0 ,0 ,0) rgb[2] = (0 ,0 ,0) rgb.write() </pre>		
算法设计 阶段三 完善功能	教师提问： 在冬天的时候，如果开窗久了会觉得冷，（加湿器开久了也会导致湿度过高）所以当温湿度条件恢复到一定数值了是不是应该再提醒我们可以关窗（或其他改变条件的操作）？ 迭代的概念： 迭代是重复反馈过程的活动，其目的通常是为了逼近所需目标或结果。每一次对过程的重复称为一次“迭代”，而每一次迭代得到的结果会作为下一次迭代的初始值。	按照阶段一中的测试方法，测试执行某一操作后多久可以关闭（或停止）。	此步作为项目收尾，留下新的任务，让学生了解项目的迭代与升级的概念。

3. 小结与点评

展示交流大家的作品，并分享自己“独特”的设计理念。

教师小结：

本节课，我们根据之前所学的数据和信息概念及其特征的内容进行了实践，亲身体验了数据、信息和知识之间的关系。并利用多种思维工具来辅助我们设计一个功能完整的作品，希望同学们能将这些思维工具用到之后的学习中，提高学习效率。

4. 延伸拓展



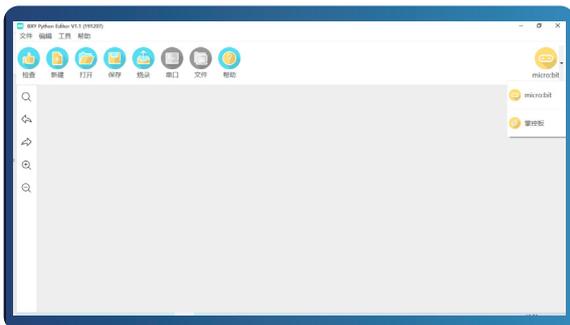
思考：

1. 关于室内环境的数据，还有什么数据可以作为室内环境健康度的参考值？我们手头是否有工具可以对其进行测量？
2. 影响人体健康状况的除了外部环境，还有就是人体的自身条件了，想一想如果想判断一个人的健康情况，可以从哪些身体数据进行分析？并思考分析这其中的数据、信息和知识之间的关系。

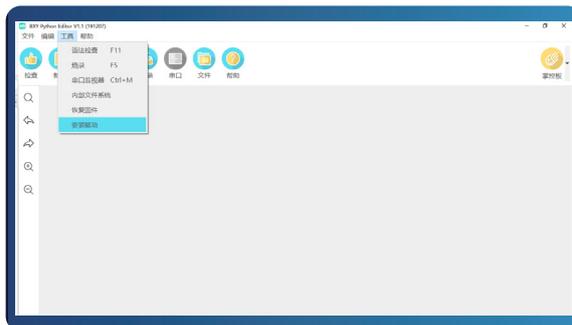
附录:

BXY 软件操作步骤

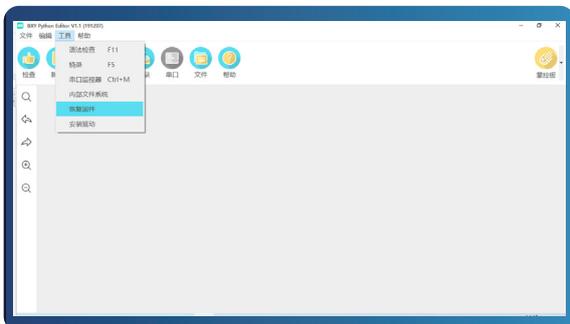
1 首先切换到掌控板模式



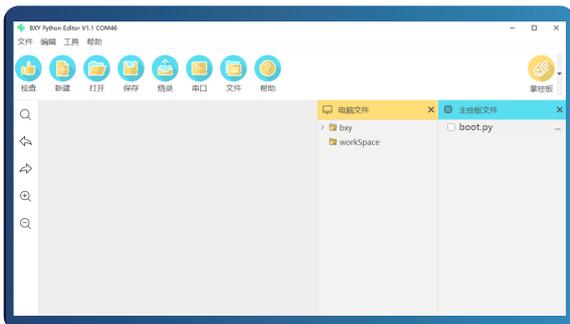
2 注意：如果是第一次在电脑上使用掌控板，需要安装驱动。



3 如果掌控板中没有固件，则会提示烧录固件，也可以手动在“工具”菜单下选择“恢复固件”。



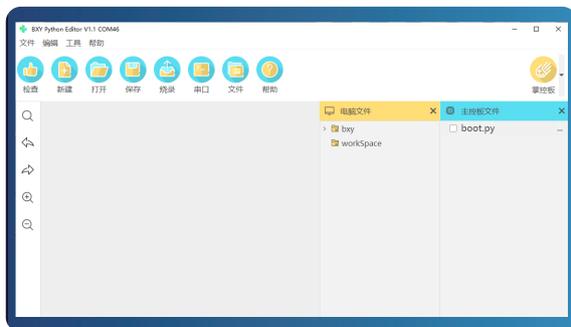
4 连接成功后“文件系统”自动弹出，包含各种示例程序以及掌控板内部的文件列表。



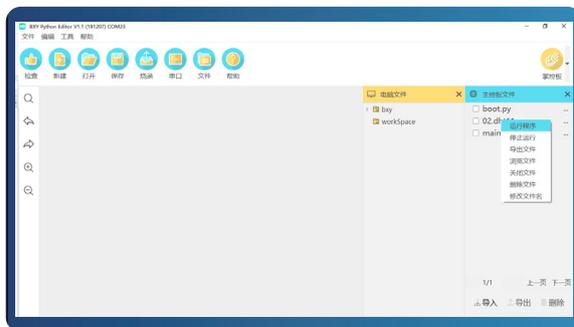
5 打开示例程序，文件 - 示例 - External-02.dht11.py（这个程序就是用于使用 dht11 传感器检测环境温湿度的示例程序）



6 点击烧录上传程序，就可以在文件系统中看到上传好的程序。



7 这时掌控板的屏幕显示内容为掌控板的 logo 及文字，需要选择对应的 py 文件右键选择“运行程序”，则掌控板一端就会开始运行。



Tips: 在 BXY 中，“烧录”操作的意义为将 python 程序上传到掌控板中，掌控板中可以存储多个 python 文件，因此需要运行程序时，可以选择对应的 py 文件右键选择“运行程序”。



注意：掌控板重启后默认运行名称为 main.py 的程序，如果需要让程序开机时自动执行则可以将文件命名为 main.py。

运行效果

在按照接线图接好模块，烧录并运行程序得到的结果如图，在屏幕上显示实时温度与湿度数据。

