

# Python 示例程序使用

文件状态:	当前版本:	V2.0
	作者:	元银萍
[ ] 正在修改	完成日期:	2019.9.4
	审核:	
[]正在发布	完成日期:	

版本历史

版本号	作者	修改日期	修改说明	审核	备注
V1.0	Adolph	2019.2.13	初始版本		
V1.1	Adolph	2019.3.7	修改路径		
V1.2	Adolph	2019.4.22	修改路径		
V2.0	元银萍	2019.9.4	更新新的		
			sensor 板相		
			关图片及修		
			改路径和运		
			行程序		



目录

前言:		 3
实验准备		 4
实验		 5
<i>—</i> `,	Led 灯实验	 5
`\	舵机实验	 7
三、	人体红外感应实验	9
四、	温湿度测试实验	 
五、	超声波测距实验	 14
六、	光敏电阻实验	 
七、	电位器实验	 
八、	蜂鸣器实验	 21
九、	点阵实验	 23
+、	数码管实验	25
+-,	按键实验	 27





本文档主要介绍虚谷号内置示例程序的测试方法和示例程序的说明。各 sensor 分 布如下图所示:



#### 上图每个标号的 Sensor 介绍如下表:

标号	Sensor 名称
1	LED 灯
2	府它机
3	人体红外传感器
4	温湿度传感器
5	超声波传感器
6	蜂鸣器
7	光敏电阻
8	电位器
9	点阵
10	数码管

	J	虚谷号实验箱
11	MIC	
12	按键	
13	喇叭	
14	耳机	

## 实验准备

1、 拿到实验箱并打开,先接通 HDMI 转接板的电源,后接通虚谷号板子的电源,等待进入系统桌面。开机后屏幕显示如下:



2、将下图所示的开关拨到指定的位置







实验

- 一、 Led 灯实验
  - 1、 实验步骤
    - 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



2) 在命令窗口中输入: python Led\_test.py 后按下 Enter 键, LED sensor 就会依据 Led\_test.py 文件的代码执行相应的操作。运行图片如下:







#### 2、 实验现象

程序运行结时 LED 灯会不停的闪烁不同的颜色,如下图所示:





## 3、 实验结束

运行结束后, LED sensor 闪烁也会停止。

#### 4、 实验代码

Led 灯测试程序代码:

from xugu import \* #导入 xugu 库 import time #导入时间模块



$pin_r = Pin(9, Pin. OUT)$	#选择 soc 控制引脚
----------------------------	--------------

pin\_g = Pin(11, Pin. OUT) #选择 soc 控制引脚

pin\_b = Pin(12, Pin. 0UT) #选择 soc 控制引脚

test = 20 #计数

1 = [0,1] #IO 口高低电平列表

while test > 0:

pin\_r.write\_digital(1[0]) #默认写入列表的第一个参数

time.sleep(1) #休眠一秒

pin\_g.write\_digital(1[0]) #默认写入列表的第一个参数

time.sleep(1) #休眠一秒

pin\_b.write\_digital(1[0]) #默认写入列表的第一个参数

1 = 1[: : -1] #列表参数交换

time.sleep(1) #休眠一秒

test-=1 #计数自减 1

## 二、 舵机实验

#### 1、 实验步骤

1) 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H) scope@localhost:~/vvBoard/Python/01.example/11.vvBox\$	-	Tannina	149世	مهامرماله	anti - And	any d/Duthan /04 avampla/44 un/Pav	_	-	V
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H) scope@localhost:~/vvBoard/Python/01.example/11.vvBox\$		iermina	i ≝\$\$/雨 - SCO	pe@iocain	IOSC: ~/VVE	soard/Pychon/01.example/11.VVBox		-	~
scope@localhost:~/vvBoard/Python/01.example/11.vvBox\$	_ 文件(F)	) 编辑(E)	) 视图(V)	终端(T)	标签 <b>(</b> A)	帮助(H)			
	scope@	localhos	t:~/wvBoa	ird/Pytho	n/01.exa	mple/11.vvBox\$			

2) 在命令窗口中输入: python Steering\_gear\_test.py 后按下 Enter 键, 舵机 sensor 就会依据 Steering\_gear\_test.py 文件的代码执行相应的操作。运行图片如下:



#### 2、 实验现象

程序运行时, 舵机会左右转动, 如下图所示:



#### 3、 实验结束

程序运行结束后, 舵机 sensor 会自动停止动作。

## 4、 实验代码

舵机测试程序代码:

```
from xugu import * #倒入 xugu 库
```

import time #倒入时间模块

```
servo = Servo(3) #选择 soc 控制引脚
```

test = 10 #计数

```
1 = [180,0] #角度列表
```

while test>0:

```
servo.write_angle(1[0]) #默认写入列表的第一个参数
```

```
1 = 1[::-1] #列表参数交换
```

```
time.sleep(1) #休眠一秒
```

```
test-=1 #计数自减1
```

## 三、人体红外感应实验

1、 实验步骤

1) 虚谷号进入桌面后,同时按住 Ctrl、Alt、t 按键,屏幕会出现一个命令窗口,在命



令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:

-	Terminal	终端 - scop	pe@localh	iost: ~/vvB	oard/Pyth	on/01.exa	mple/11.vvB	lox	-	÷.	×
文件(F	) 编辑(E)	视图(V)	终端(T)	标签(A)	帮助(H)						
scope@	localhosi	t∶~∕wvBoa	rd/Pytho	n/01.exa	mple/11.v	vBox\$					
						_					

 2) 在命令窗口中输入: python Human\_body\_induction\_test.py 后按下 Enter 键, 人体 红外 sensor 就

会依据 Human\_body\_induction\_test.py 文件的代码执行相应的操作。运行图片如下:





```
- + \times
 ÷.
       Terminal 终端 - scope@localhost: ~/vvBoard/Python/01.example/11.vvBox
 文件(F)
         编辑(E)
                  视图(V) 终端(T) 标签(A)
                                             帮助(H)
scope@localhost:~/vvBoard/Python/01.example/11.vvBox$ python Human body inductio
n test.py
pymata aio Version 2.28 Copyright (c) 2015-2018 Alan Yorinks All rights reserved
Using COM Port:/dev/ttyS1
Initializing Arduino - Please wait...
Arduino Firmware ID: 2.5 StandardFirmata.ino
Auto-discovery complete. Found 20 Digital Pins and 6 Analog Pins
find human!!!
no human!!!
find human!!!
no human!!!
find human!!!
no human!!!
no human!!!
find human!!!
find human!!!
find human!!!
```

#### 2、 实验现象

程序运行时,当人体感应传感器检测到有人动时会显示 find human!!!, 当没有检测 到人动的时候会显示 no human!!!。

#### 3、 实验结束

程序运行结束后,人体红外 sensor 会停止工作。

#### 4、 实验代码

人体红外测试程序代码:

from xugu import Pin #从 xugu 库中导入 Pin 类

import time #导入时间模块

pin = Pin(4,Pin.IN) #创建Pin 对象

test = 30

while test>0:

value = pin.read\_digital() #读取对应 Pin 脚的电平值

if value:

print("find human!!!") #电平为1代表发现人

else:

print("no human!!!") #电平为0代表没发现人



四、 温湿度测试实验

#### 1、 实验步骤

 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/sensor/回车会出现如下 图所示的命令框:

	Terminal	终端 - scop	e@localh	ost: ~/vvB	oard/Python/01.example/11.vvBox	-	+	$\times$
文件(F	) 编辑(E)	视图(V)	终端(T)	标签 <b>(</b> A)	帮助(H)			
scope@	localhost	:∼/vvBoa	rd/Pytho	n/01.exa	mple/11.vvBox\$			
								U

2) 在命令窗口中输入: python Temperature\_and\_humidity\_test.py 后按下 Enter 键, 温湿度传感器就会依据 Temperature\_and\_humidity\_test.py 文件的代码读取温湿度的值将显示在终端上,前面的值为湿度,后面的值为温度。运行图片如下:



```
+ \times
       Terminal 终端 - scope@localhost: ~/vvBoard/Python/01.example/11.vvBox
 Ŧ
 文件(F)
        编辑(E) 视图(V) 终端(T) 标签(A)
                                           帮助(H)
avrdude: verifying ...
avrdude: 8292 bytes of flash verified
avrdude: safemode: hfuse reads as 0
avrdude: safemode: efuse reads as 0
avrdude: safemode: Fuses OK (E:00, H:00, L:00)
avrdude done. Thank you.
burn complete
-1
-1
77.0029.00
70.0029.00
70.0029.00
70.0029.00
70.0029.00
70.0029.00
-1
70.0029.00
70.0029.00
```

#### 2、 实验结果

程序运行时,我们对着温湿度传感器吹气,传感器会上报不同的温度和湿度上来,并显示 在终端上;如下图的两个值,没吹气湿度是 68%,温度是 29度;吹气后,湿度是 79%,温 度是 32度。



3、 实验结束

需要结束实验时,同时按下键盘的 Ctrl 和 C 按键结束程序

## 4、 实验代码

温湿度测试程序代码:

from dhtc import DHT #从 dhtc 库中导入 DHT 模块

import time #导入时间模块

dht = DHT() #创建 DHT 对象

test = 12 #计数

#### while test >0: #创建一个循环



t =dht.read() #将读取到的温湿度的值保存在变量 t 中
 print(t) #将温湿度的值打印到终端
 time.sleep(2.5) #睡眠 2.5 秒
 test -= 1 #计数自减 1

## 五、超声波测距实验

#### 1、 实验步骤

1) 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



2) 超声波接入 sensor 板示意图如下:







3) 在命令窗口中输入: python Ultrasonic\_test.py 后按下 Enter 键,温湿度传感器就 会依据 Ultrasonic\_test.py 文件的代码读取温湿度的值将显示在终端上。运行图片 如下:



#### 2、 实验现象

程序运行时,通过移动超声波传感器,可以获取不同的距离,当传感器贴近障碍物时,距离显示为-1,其他显示对应的距离值。

#### 3、 实验结束





程序读取 30 次距离值后自动退出。

4、 实验代码

超声波测试程序代码:

from dhtc import HC #从 dhtc 模块中导入 HC 类

import time #导入时间模块

hc = HC() #创建 HC 对象

test = 30 #计数

while test>0:

1 = hc.read() #读取距离

print(1) #在终端上输出距离

time.sleep(1) #睡眠一秒

test-=1 #计数自减 1

- 六、 光敏电阻实验
  - 1、 实验步骤
    - 1) 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



-	[erminal #	终端 - scor	e@localh	ost: ~/vvB	oard/Python/01.example/11.vvBox	- + >	<
文件(F)	编辑(E)	视图(V)	终端(T)	标签(A)	帮助(H)		
scope@l	ocalhost	:~/vvBoa	rd/Pytho	n/01.exa	mple/11.vvBox\$		

2) 在命令窗口中输入: python LightSensor\_test.py 后按下 Enter 键,程序就会依据 LightSensor\_test.py 文件的代码从 arduino 的 A1 模拟输入口读取对应的值。运行图 片如下:

▼ Terminal 终端 - scope@localhost: ~/vvBoard/Python/01.example/11.vvBox		+ x
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)		
Using COM Port:/dev/ttyS1	i i	
Initializing Arduino - Please wait Arduino Firmware ID: 2.5 StandardFirmata.ino Auto-discovery complete. Found 20 Digital Pins and 6 Analog Pins		
160 225 760 159 159 159 159		
161 551 159 160 345 160 160 167		

2、 实验现象

程序运行时,当我们没有遮挡光敏电阻时,读出 A1 脚在当前的亮度下的值是 160 左右;



当我们用东西遮挡住光敏电阻时, A1 脚的读数就会上升到 551 左右, 这时 LED 灯绿灯将 被点亮, 如下图所示:

光敏电阻无遮挡现象







3、 实验结束

程序在 30 次计数结束后自动退出。

#### 4、 实验代码

光敏电阻测试程序代码

from xugu import Pin #从 xugu 库中导入 Pin 类

import time #导入时间模块

lightSensor\_pin = "a1" #定义模拟输入引脚

led\_pin = 11 #定义 LED 控制引脚

lightSensor = Pin(lightSensor\_pin, Pin. ANALOG) #创建 Pin 对象

led = Pin(led\_pin, Pin.OUT) #创建 Pin 对象

test = 30 #计数

while test > 0:

value = lightSensor.read\_analog() #读取 A1 脚的模拟输入值

print(value)

if value > 200: #判断 A1 脚获取的值是否大于 200

led.write\_digital(1) #点亮绿色的 LED 灯

else:

led.write\_digital(0) #熄灭 LED 灯

time.sleep(1) #睡眠1秒

test -= 1 #计数自减 1





- 七、电位器实验
  - 1、 实验步骤
    - 1) 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



2) 在命令窗口中输入: python Potentiometer\_test. py 按下 Enter 键后,程序就会依据 Potentiometer\_test. py 文件的代码从 arduino 的 A0 模拟输入口读取对应的值。运行 图片如下:



	т т	erminal <b>≜</b>	冬端 - scop	e@localh	ost: ~/vvB	oard/Python/01.example/11.vvBox — + >	<
	文件(F)	编辑(E)	视图(V)	终端(T)	标签(A)	帮助(H)	
	Initiali	zing Arc	luino - F	lease w	ait		
	Arduino	Firmware	e ID: 2.5	5 Standa Found	rd⊦ırmata 20 Digita	).100 D Dins and 6 Analog Dins	
	AULO-UIS	covery c	.omptere.	. rouniu .	20 DIGIC	IL FINS AND O ANALOG FINS	
							-
	302						
	300						
	307						
	302						
	305						
	303						
	305						
	315						
	33/ ece						
	1023						
	478						
	479						
	478						
	481						
	483						
. 1							

## 2、 实验现象

程序运行时,我们拨动电位器到如下图片的位置时



我们终端将获得对应的值,分别是0、415和1023

#### 3、 实验结束

程序在 30 次计数结束后自动退出。

#### 4、 实验代码

电位器测试程序代码

from xugu import Pin #从 xugu库中导入 Pin类

import time #导入时间模块

potentiometer\_pin = "a0" #定义模拟输入引脚

potentiometer = Pin(potentiometer\_pin,Pin.ANALOG) #创建Pin 对象

test = 30 #计数

while test > 0:



value = 1023-potentiometer.read\_analog() #读取 A0 脚的模拟输入值,因电位器

接反,需要使用 1023 减去获取的值

value = potentiometer.read\_analog() #读取 A0 脚的模拟输入值

print(value) #将读取的值打印到终端

time.sleep(1) #睡眠1秒

test -= 1 #计数自减 1

## 八、蜂鸣器实验

- 1、 实验步骤
  - 1) 虚谷号进入桌面后,同时按住 Ctrl、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



2) 在命令窗口中输入: python Buzzer\_test.py 后按下 Enter 键,蜂鸣器就会依据 Buzzer\_test.py 文件的代码发出声音。运行图片如下:



▼ Terminal 终端 - scope@localhost: ~/vvBoard/Python/01.example/11.vvBox = + × 文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H) scope@localhost: ~/vvBoard/Python/01.example/11.vvBox\$ python Buzzer\_test.py pymata\_aio Version 2.28 Copyright (c) 2015-2018 Alan Yorinks All rights reserved . Using COM Port:/dev/ttyS1 Initializing Arduino - Please wait... Arduino Firmware ID: 2.5 StandardFirmata.ino Auto-discovery complete. Found 20 Digital Pins and 6 Analog Pins Task was destroyed but it is pending! task: <Task pending coro=<PymataCore.\_command\_dispatcher() running at /usr/local /lib/python3.5/dist-packages/pymata\_aio/pymata\_core.py:1400> wait\_for=<Future fi nished result=None>> scope@localhost:~/vvBoard/Python/01.example/11.vvBox\$

#### 2、 实验现象

程序运行时,我们可以听到蜂鸣器发出滴滴滴的声音

#### 3、 实验结束

蜂鸣器响 30 声后自动退出。

#### 4、 实验代码

蜂鸣器测试程序代码

from xugu import \* #导入 xugu 库

import time #导入时间模块

buzzer = Pin(2,Pin.OUT) #创建Pin 对象

test = 30 #计数

while test > 0:

buzzer.write\_digital(1) #拉高 soc 的 12 脚

time.sleep(0.1) #睡眠 100 毫秒

buzzer.write\_digital(0) #拉低 soc 的 12 脚

time.sleep(0.1)

test -= 1 #计数自减1



buzzer.write\_digital(0) #拉低 soc 的 12 脚

- 九、 点阵实验
  - 1、 实验步骤

1) 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



2) 在命令窗口中输入: python Matrix\_test.py 按下 Enter 键后,程序就会依据 Matrix\_test.py 文件的代码向点阵芯片写入对应的值。运行图片如下:



 $+ \times$ π. Terminal 终端 - scope@localhost: ~/vvBoard/Python/01.example/11.vvBox 文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H) avrdude: 7866 bytes of flash written avrdude: verifying flash memory against /home/scope/software/resource/matrix\_8x8 /matrix\_8x8.ino.hex: avrdude: load data flash data from input file /home/scope/software/resource/matr ix\_8x8/matrix\_8x8.ino.hex: avrdude: input file /home/scope/software/resource/matrix\_8x8/matrix\_8x8.ino.hex contains 7866 bytes avrdude: reading on-chip flash data: avrdude: verifying ... avrdude: 7866 bytes of flash verified avrdude: safemode: hfuse reads as 0 avrdude: safemode: efuse reads as 0 avrdude: safemode: Fuses OK (E:00, H:00, L:00) avrdude done. Thank you. burn complete

#### 2、 实验现象

程序运行时,点阵芯片会显示对应的字符,如下图所示:



#### 3、 实验结束

需要结束实验时,同时按下键盘的Ctrl和C按键结束程序

4、 实验代码

from mtx import MTX #从 mtx 库中导入 MTX 类
import time #导入时间模块
mtx = MTX() #创建 MTX 类
val = "123abc" #将需要显示的值赋值给 val



while True: #创建循环

if mtx.burn\_complete: #判断 arduino 驱动是否烧入完毕

mtx.write(val) #写入 val 的值

time.sleep(6) #休眠6秒

## 十、数码管实验

- 1、 实验步骤
  - 1) 虚谷号进入桌面后,同时按住 Ctr1、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01.example/11.vvBox/回车会出现如下图所示的 命令框:



2) 在命令窗口中输入: python Nixietube\_test.py 按下 Enter 键后,程序就会依据 Nixietube\_test.py 文件的代码向数码管芯片写入对应的值。运行图片如下:





#### 2、 实验现象

程序运行时,数码管先从左边第0位数码管从0<sup>~</sup>9点亮,之后是点亮第1位数码管,以此类推,现象如下图片所示:



## 3、 实验结束

需要结束程序时,同时按下键盘的Ctrl和C按键结束程序

4、 实验代码

from tm1637 import TM1637 #从 tm1637 库中导入 TM1637 类 import time #导入时间模块 tm1637 = TM1637() #创建 TM1637 类 val = "0123456789" #将数码管需要显示的值赋值给 val com = "0123" #将数码管的位数赋值给 com



#### i = 0 #数码管位数计数

j = 0 #显示内容计数

tm1637.clearDisplay() #数码管清除显示

while True: #创建循环

for letter in com: #遍历 com 字符串

for letter in val: #便利 val 字符串

tm1637.display(com[i], val[j]) # 数码管的第 i 位显示字符串的第 j 数

字

time.sleep(1) #休眠1秒

j += 1 #j计数加1

i += 1 #i 计数加 1

j = 0 #将 j 计数重新赋值为 0

i = 0 #将 i 计数重新赋值为 0

tm1637.clearDisplay() #数码管清除显示

time.sleep(1) #休眠1秒

## 十一、 按键实验

#### 1、 实验步骤

1) 虚谷号进入桌面后,同时按住 Ctrl、Alt、t 按键,屏幕会出现一个命令窗口,在命 令窗口中输入: cd vvBoard/Python/01. example/11. vvBox/回车会出现如下图所示的命令 框:



٦ 💌	[erminal #	终端 - scop	e@localh	ost: ~/vvB	oard/Python/01.example/11.vvBox	-	÷	×
文件(F)	编辑(E)	视图(V)	终端(T)	标签 <b>(</b> A)	帮助(H)			
scope@l	ocalhost	:~/vvBoa	rd/Pytho	n/01.exar	mple/11.vvBox\$			

2) 在命令窗口中输入: python Button\_test.py 按下 Enter 键后,程序就会依据 Button\_test.py 文件的代码向数码管芯片写入对应的值。运行图片如下:



#### 2、 实验现象

程序运行时,不按按键终端显示 not press key,按下上方的按键,终端显示 Up,按下下 方的按键,终端显示 Down,按下中间的按键,终端显示 Middle,按下左边的按键,终端



显示 Left,按下右边的按键,终端显示 Right。

## 3、 实验结束

需要结束程序时,同时按下键盘的 Ctrl 和 C 按键结束程序

## 4、 实验代码

from xugu import Pin # 从 xugu 库中导入 Pin 类
import time
p = Pin("A2", Pin.ANALOG) # 初始化 A2 引脚,设置为输入模式
key = -1
while True:
value = p.read_analog() #读取 A2 引脚的电压值
if value != key: #去抖
time.sleep(0.05)
<pre>value = p. read_analog()</pre>
if value != key:
key = value
if value >= 0 and value <= 100:
print("Up")
if value $>$ 100 and value $\leq$ 300:
<pre>print("Right")</pre>
if value $>$ 300 and value $\leq$ 600:
print("Down")
if value $>$ 600 and value $\leq$ 800:
<pre>print("Left")</pre>
if value $>$ 800 and value $\leq$ = 1000:
<pre>print("Middle")</pre>
if value > 1000:
print ("not press key")