

# 虚谷号GPIO范例

GPIO（英语：General-purpose input/output），通用型之输入输出的简称，可以理解为对芯片引脚电平的编程控制和状态读取。

为降低初学者的开发门槛，虚谷号给Arduino UNO烧写了标准的Firmata协议，通过串口命令进行控制Arduino引脚，并提供了和MicroPython语法完全兼容的xugu库，供Python编程教学或者作品开发时调用。

## 1.数字输出

数字信号的输出。只有两种状态，0（低电平）和1（高电平），调用Pin.write\_digital (value) 函数，给引脚设置电平值。value 指要设置的电平值,1 代表高电平,0 代表低电平。该方法在输出模式有效。

### (1) 调用xugu库中Pin类

引用xugu库时，虚谷号将检查板载Arduino上是否烧写了Firmata协议。如果没有，将调用烧写工具，给Arduino写Firmata协议的固件。这个过程需要十多秒时间。你将看到板子上有Led在闪烁。

In [ ]:

```
from xugu import Pin
```

选择引脚初始化

In [ ]:

```
p = Pin(13, Pin.OUT)
#等价的写法: p = Pin("D13", Pin.OUT)
```

设置为1（高电平），点亮13脚的LED灯

In [ ]:

```
# 设置 13号引脚为高电平
p.write_digital(1)
```

设置为0（低电平），熄灭13脚的LED灯

In [ ]:

```
p.write_digital(0) # 设置 13 号引脚为低电平
```

注：运行到这里，如果需要运行下面的程序或者关闭窗口，请先点击上方的黑色正方形中断服务。

## 范例：虚谷闪一闪1

让虚谷号自带的LED(接在13号引脚，即D13)闪烁，设置为亮1秒熄灭1秒。

采用数字输出的方式，使用xugu库的Pin类和时间模块进行，代码如下：

In [ ]:

```
import time # 导入 time 模块
from xugu import Pin # 从 xugu 库中导入 Pin 类
led = Pin(13, Pin.OUT) # 初始化 Pin 类
# 等价的写法:led = Pin("D13", pin.OUT)
while True:
    # 用循环实现持续地开灯关灯,到达闪烁的效果
    led.write_digital(1) # 点亮连接13号引脚的 LED 灯
    time.sleep(1) # 持续 1 秒
    led.write_digital(0) # 关闭 LED 灯
    time.sleep(1) # 持续 1 秒
```

注：运行到这里，如果需要运行下面的程序或者关闭窗口，请先点击上方的黑色正方形中断服务。

(2) 调用xugu库中Pin类

xugu库中还有一个特殊的类：LED。LED类使用方法和Pin类中的数字输出很像。

## 范例：虚谷闪一闪2

让虚谷号自带的LED（接在13号引脚,即D13）闪烁，设置为亮2秒熄灭3秒。

使用xugu库的LED类和时间模块进行，代码如下：

In [ ]:

```
import time #导入time模块
from xugu import LED #从xugu库中导入LED类
led = LED(13) # 初始化LED类

while True: # 用循环实现持续地开灯关灯，到达闪烁的效果
    led.on() # 点亮连接13号引脚的LED灯
    time.sleep(2) # 持续2秒
    led.off() # 关闭LED灯
    time.sleep(3) # 持续3秒
```

注：运行到这里，如果需要运行下面的程序或者关闭窗口，请先点击上方的黑色正方形中断服务。

## 2.数字输入

输入数字信号，调用Pin.read\_digital()函数，返回该IO引脚电平值，1代表高电平，0代表低电平。该方法在输入模式有效。

调用xugu库的Pin类和时间模块，读取D4引脚的高低电平状态，并打印获取的值。

## 范例：定时读取数字传感器

让虚谷号定时读取数字传感器的值，并打印输出，数字传感器接在4号引脚。

使用xugu库的Pin类和时间模块进行，代码如下：

In [ ]:

```
from xugu import Pin # 从 xugu 库中导入 Pin类
import time # 导入时间模块
```

选择引脚初始化。

In [ ]:

```
p = Pin(4, Pin.IN)
```

每100ms读取一次值并打印结果。

In [ ]:

```
while True:
    value=p.read_digital() # 读取 4 号引脚电平信息,赋值给 value
    print(value) # 将value的值打印到终端上
    time.sleep(0.1) # 持续100ms
```

常见的数字传感器有按钮、触摸、红外测障等。

注：运行到这里，如果需要运行下面的程序或者关闭窗口，请先点击上方的黑色正方形中断服务。

### 3.模拟输出

直接输出模拟信号，调用Pin.write\_analog(value)函数，给引脚设置模拟值。value指要设置的模拟值,数据范围在0和255之间。该方法在输出模式有效。虚谷号支持的模拟输出(PWM 方式)的引脚共有6个：3、5、6、9、10、11。

#### 范例：PWM方式控制LED

用PWM方式控制LED模块，接在虚谷号的10号引脚上。

单片机的模拟电平输出，一般都是采用PWM脉宽调制实现，Arduino上支持PWM的引脚有3，5，6，9，10，11，在虚谷号板子上可以看到引脚旁边有“~”的标志。

使用xugu库的Pin类，代码如下：

In [ ]:

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

选择引脚初始化

In [ ]:

```
p = Pin(10, Pin.OUT)
```

给引脚设置模拟值128

In [ ]:

```
p.write_analog(128)
```

给引脚设置模拟值255

In [ ]:

```
p.write_analog(255)
```

小挑战：写一个循环语句，控制LED的明暗交替变化，看看能不能实现呼吸灯的效果。

In [ ]:

```
import time # 导入时间模块
v=5
for i in range(1000):
    if i>255:
        v=v*-1
    p.write_analog(i)
    i=i+v
    time.sleep(0.1) # 持续100ms
```

## 4.模拟输入

输入模拟信号，调用Pin.read\_analog()函数，返回IO引脚的模拟值,数据范围在0和1023之间。该方法在输入模式有效。

### 范例：定时读取模拟传感器

读取虚谷号A0引脚的电压值并打印，A0引脚接模拟传感器。常见的模拟传感器有旋钮、光线、声音、温度等。

使用xugu库的Pin类，代码如下：

In [ ]:

```
from xugu import Pin
```

选择引脚初始化

In [ ]:

```
p = Pin("A0", Pin.ANALOG)
```

读取 A0 引脚的电压值。

In [ ]:

```
value = p.read_analog()
```

打印获取的电压值。

```
In [ ]:
```

```
print(value)
```

小挑战：写一个循环语句，定时读取模拟传感器的数值并打印。

```
In [ ]:
```

```
import time # 导入时间模块
while True:
    value=p.read_analog() # 读取 A0 引脚电平信息,赋值给 value
    print(value) # 将value的值打印到终端上
    time.sleep(0.1) # 持续100ms
```

注：运行到这里，如果需要运行下面的程序或者关闭窗口，请先点击上方的黑色正方形中断服务。

## 5.舵机控制

控制舵机旋转的角度，调用Servo(pin\_num)函数，pin\_num 引脚标号,可以直接传入虚谷板上的引脚编号,例如 D3 或者 A5,也可以直接传入 13或者 19 这样的数字。还调用Servo.write\_angle(value)函数，让舵机转动到指定角度,value 指角度。每种舵机的最大转动角度不一样,需要参考舵机说明书。

### 范例：控制舵机旋转

使用xugu库中Servo类，控制D4引脚上的舵机旋转。

```
In [ ]:
```

```
from xugu import Servo
```

选择控制的引脚初始化并连接到舵机

```
In [ ]:
```

```
servo = Servo(4)
```

设置值为150,舵机旋转150°

```
In [ ]:
```

```
servo.write_angle(150)
```

注：运行到这里，如果需要运行下面的程序或者关闭窗口，请先点击上方的黑色正方形中断服务。

## 6.串口通讯

将 Arduino 的串口通信内容转发到 PC 的串口。调用SerialMgt(port, baudrate)函数, port 指虚谷号连接 pc 的串口号,baudrate 为串口波特率。当不设置串口号和波特率时,虚谷库默认使用/dev/ttyGS0 串口,波特率为 115200。还调用SerialMgt.write(data)向串口中写入数据,data 为写入的数据,类型为 String。

使用xugu 库中Pin、XuguLog、SerialMgt类, 读取模拟传感器数值打印到文本内并通过 PC 串口输出。



注意: 使用串口功能, PC端要安装串口驱动软件, 在vvBoard文件夹中。

In [ ]:

```
from xugu import Pin, XuguLog, SerialMgt # 从 xugu 库中导入 Pin、XuguLog、SerialMgt类
import time # 导入时间模块
```

设置读取模拟传感器数值的次数

In [ ]:

```
test = 20
```

选择读取的引脚初始化

In [ ]:

```
p = Pin("A0", Pin.ANALOG)
```

初始化串口

In [ ]:

```
ser = SerialMgt()
```

创建文本保存的路径。不指定路径时, 文件保存在与虚谷号GPIO范例同级路径下

In [ ]:

```
f=XuguLog('read_analog_20_times.log') # 当不指定路径时,log文件保存在与虚谷号GPIO范例同级路径下
#f=XuguLog('/home/scope/analog_input.log') # 初始化日志对象,当指定路径时 log 文件保存在指定路径下
```

注: 如果使用U盘模式运行使用XuguLog功能的代码, 将无法获取log文件!!! 需要指定路径 (/home/scope/vvBoard/Python)

每500ms读取数值保存在文本内并写入到串口, 循环读取20次, 在 PC 端建议通过串口工具打开对应的串口号, 即可查看虚谷板写入的内容。

In [ ]:

```
while test > 0:
    value = p.read_analog() #读取 A0 引脚的电压值
    f.write(str(value)) # 将变量 value 写入到日志文件中
    print(value) # 将 value 的值打印到终端
    ser.write(str(value).encode()) # 将 value 的值写入串口
    test -= 1 # 计数自减 1
    time.sleep(0.5) # 持续 500ms
```

注：运行到这里，如果需要关闭窗口，请先点击上方的黑色正方形中断服务，否则代码还会在持续运行。如果打开笔记的时候，程序已经运行，需先选择上方“服务”->再选择“重启 & 清空输出”，便可重新运行程序

In [ ]: