虚谷号xugu库（基于Python的GPIO控制）使用文档

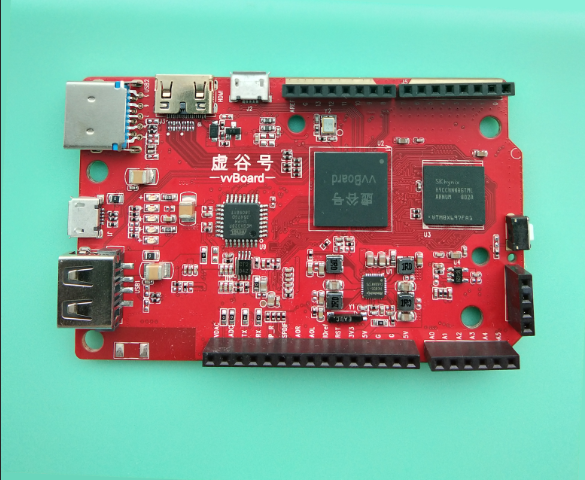
——基于人民教育出版社和中国地图出版社教材

第一部分、虚谷号GPIO简介

1.虚谷号简介

虚谷号是一个面向人工智能教学和Python编程学习的中国原创开源硬件平台，板内集成高性能处理器和通用单片机，内置多功能扩展接口和多种通信接口，为人工智能和Python编程教学提供完整课程资源包。

虚谷号运行完整的Linux系统，有HDMI接口，2个USB3.0，自带蓝牙和Wi-Fi功能，接上鼠标键盘和显示器，就是一台完整的电脑。虚谷号板载了ATMEGA 328P芯片，引脚资源完全兼容Arduino（可以之间看成是一块Arduino UNO），其GPIO功能主要是通过这块328P芯片来完成。



2.GPIO功能简介

虚谷号提供了多种方式，实现GPIO功能。

方式1：虚谷号通过串口和板载的Arduino UNO连接，可以用任何一款Arduino的编程工具，用标准的Arduino代码进行编写，控制Arduino UNO的所有引脚。

PC

虚谷号

核心系统

板载Arduino

标准Arduino程序

方式2：虚谷号给Arduino UNO写上标准的FirMata协议，通过串口命令进行控制Arduino引脚，为降低初学者的开发门槛，系统提供了和MicroPython语法完全兼容的“xugu库”供Python编程时调用。

PC

虚谷号

核心系统

板载Arduino

FirMata协议

方式3：虚谷号的主芯片引出I2C接口，可以通过这一接口来实现对外部设备的控制。

PC

虚谷号

核心系统

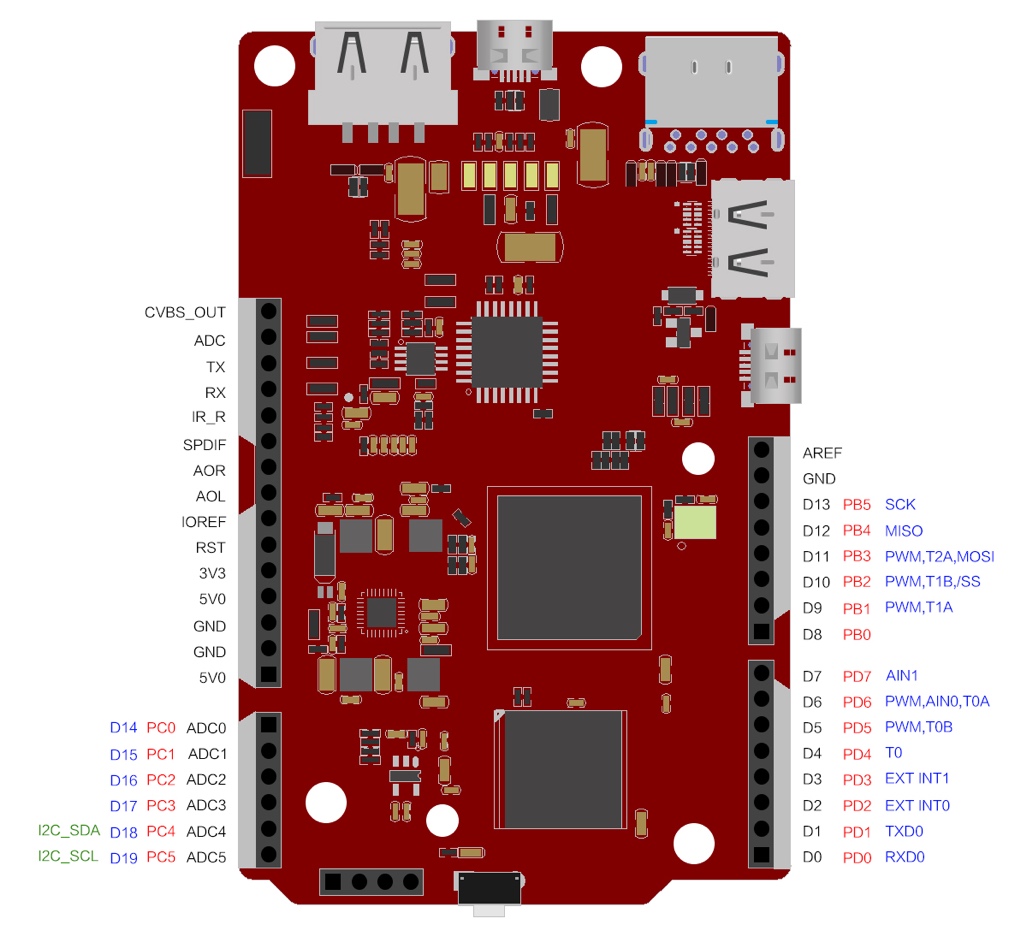
I2C设备

I2C接口

第三种方式中，虚谷号主芯片的I2C接口可以和方式1、方式2结合。相对来说，方式2的开发难度最低，只需要有Python编程基础即可。**本文档介绍的是以方式2和方式3为主，即介绍虚谷号定制的Python库（xugu）的使用。**

|  |  |  |  |
| --- | --- | --- | --- |
| **对比项** | **开发难度** | **开发基础** | **功能** |
| 方式1 | 较高 | Ardunio、python | 兼容 |
| 方式2 | 较低 | python | 基础输入输出 |
| 方式3 | 较低 | python |  |

3.引脚说明图示



第二部分、推荐编程工具

1.Vscode

2.uPyCraft

3.mPython

第三部分、xugu库简介

xugu库中有Pin、ADC··、XuguLog、S2W等类。

1.控制I/O引脚：Pin类

Pin对象用于控制I/O引脚，通常与可以驱动输出电压和读取输入电压的物理引脚相关联。Pin类具有设置引脚模式（IN，OUT等）的方法以及获取和设置数字逻辑电平的方法。有关引脚的模拟控制，请参阅ADC类。

Pin类提供了多种方式对引脚的模式进行定义。引脚分为两种状态：输出（OUT）和输入（IN）以及三种信号类型：数字信号（DIGITAL）、模拟信号（ANALOG）和脉冲信号（PWM）。

*\_\_init\_\_(pin\_num, pin\_model, pin\_type=None, port="/dev/ttyACM0",debug=False）*

pin\_num传入针脚标号，可以直接传入虚谷板上的针脚编号，例如D3或者A5，也可以直接传入13或者19这样的数字。如果不指定pin\_type, 将会根据传入的针脚编号设置成对应的模式。例如会将D3针脚设置为数字信号，A5设置为模拟信号，将15设置为模拟信号，将13设置为数字信号。

pin\_model 针脚模式，IN是输入模式，OUT是输出模式

pin\_type 针脚类型，DIGITAL为数字信号，ANALOG为模拟信号，PWM为脉冲信号

port 串口号，虚谷板中默认为/dev/ttyACM0，不建议修改该值

debug 调试模式，默认为False, 如果设为True, 将会在日志里看到调试信息

*high()*

给针脚一个高电位，只有在输入模式有效，当该针脚接入LED灯的时候，灯会点亮

*low()*

给针脚一个低电位，只有在输入模式有效，当该针脚接入LED灯的时候，灯会熄灭

*on()*

等价与high()

*off()*

等价与low()

*value()*

读取针脚的电平值，只有在输入模式有效

2.模拟信号转数字信号：ADC类

该类能将未知的模拟信号转换成为可被描述的已知信号

*\_\_init\_\_(pin)*

pin Pin对象，Pin对象模式为输入模式，类型为模拟信号

*read()*

读取模拟信号转换成的数字信号

3.数字信号转模拟信号: DAC类

该类能将已知的信号转换为等效的模拟数值

*\_\_init\_\_(pin)*

pin Pin对象，Pin对象模式为输出模式，类型为模拟信号

*write(value)*

value: 已知的数字信号，将数字信号转换成等效的未知信号

4. 舵机控制: Servo类

该类可以控制舵机旋转

\_\_init\_\_(pin\_num, port="/dev/ttyACM0", debug=False)

pin\_num 针脚标号，可以直接传入虚谷板上的针脚编号，例如D3或者A5，也可以直接传入13或者19这样的数字。如果不指定pin\_type, 将会根据传入的针脚编号设置成对应的模式。例如会将D3针脚设置为数字信号，A5设置为模拟信号，将15设置为模拟信号，将13设置为数字信号。

angle(angle)

angle 转动角度，让舵机转动到指定角度。每种舵机的最大转动角度不一样。需要参考舵机说明书。

speed()

使舵机持续转动。

5. XuguLog: 日志输出类

\_\_init\_\_()

初始化该类的时候，会自动生成一个与python程序同名的日志文件，后缀为.log。每次启动python程序，会覆盖旧的日志文件，程序运行过程中，会将打印的日志信息追加到日志文件中。

write(value)

将日志信息写入到日志文件中

6. S2W: 虚谷板中sqlite数据库对象

这个类提供两个静态方法，在虚谷板中创建一个sqlte数据库表，表结构为pin\_num(文本类型)，data(整数类型)，pin\_num为主键，不能出现重复的值。

@staticmethod

write\_data(pin\_num, value)

*调用该方法时不需要初始化S2W类，可以直接调用，例如S2W.write\_data(A2, 110)*

将针脚上读取的数值保存到数据库中

pin\_num: 针脚标号，可以直接传入虚谷板上的针脚编号，例如D3或者A5，也可以直接传入13或者19这样的数字。在数据库中保存的都是统一的数字编号，例如，传入的是D3和3是等效的，传入A0和14是等效的。

value: 保存的电位值。

@staticmethod

read\_value(pin\_num)

*调用该方法时不需要初始化S2W类，可以直接调用，例如S2W.read\_value(A2)*

从数据库中读取针脚的值

pin\_num: 针脚标号，可以直接传入虚谷板上的针脚编号，例如D3或者A5，也可以直接传入13或者19这样的数字。在数据库中保存的都是统一的数字编号，例如，传入的是D3和3是等效的，传入A0和14是等效的。

第四部分、常见范例代码

**1.闪烁灯（blink.py）**

代码说明：让虚谷号自带的LED（接在13号引脚，即D13）闪烁

代码范例：

|  |
| --- |
| iimport 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，13号引脚设置为输出模式并标记为led  # 等价的写法：led = Pin(“D13”, pin.OUT)  while 1: # 做一个循环，持续的开灯关灯，到达闪烁的效果  led.on() # 点亮连接13号引脚的LED灯  # 等价的写法：led.high()  time.sleep(1) # 持续1秒  led.off() # 关闭LED灯  # 等价的写法：led.low()  time.sleep(1) # 持续1秒 |

**2.数字输出**

代码说明：将D10引脚置于高电平

代码范例：

|  |
| --- |
| from xugu import Pin # 从xugu库中导入Pin类  p = Pin(10, Pin.OUT) # 初始化Pin类，将10号引脚设置为输出模式  # p = Pin(10, Pin.OUT)  p.value(1) # 设置10号引脚为高电平  # 等价的写法：p.on() 或者 p.high() |

代码说明：将D10引脚置于低电平

代码范例：

|  |
| --- |
| from xugu import Pin # 从xugu库中导入Pin类  p = Pin(10, Pin.OUT) # 初始化Pin类，将10号引脚设置为输出模式  # p = Pin(10, Pin.OUT)  p.value(0) # 设置10号引脚为低电平  # 等价的写法：p.off() 或者 p.low() |

**3.模拟输出**

代码说明：将D3引脚设置为模拟输出（PWM），参数为0-255。

代码范例：

|  |
| --- |
| from xugu import Pin # 从xugu库中导入Pin类  p = Pin(3, Pin.PWM) # 初始化3号引脚为模拟输出模式  # 等价的写法 p = Pin(3, Pin.OUT，“A”)  # 等价的写法 p = Pin(3, Pin.OUT，Pin.ANALOG)  p.write(128) # 给引脚设置电位值128 |

注：虚谷号支持的PWM输出的引脚共有6个： 3、5、6、9、10、11。

**4.数字输入**

代码说明：读取D2引脚的高低电平状态。

代码范例：

|  |
| --- |
| from xugu import Pin, SerialMgt # 从xugu库中导入Pin、SerialMgt类  p = Pin(2, Pin.IN) # 初始化2号引脚，设置为数字输入模式  value=p.value() # 读取2号引脚电平信息，赋值给value  print(value） #在虚谷界面中打印变量value  ser = SerialMgt() # 初始化串口对象  ser.write(value) # 将变量value写入到串口中 |

**5、模拟输入**

代码说明：读取A0引脚的电位值

代码范例：

|  |
| --- |
| from xugu import Pin, ADC，SerialMgt # 从xugu库中导入Pin、ADC、SerialMgt类  p = Pin("A0") # 初始化A0引脚，设置为输入模式  # 等价的写法 p = pin("A0",Pin.IN,Pin.ANALOG)  adc = ADC(p) # 初始化模拟信号对象  print(adc.value()) # 在虚谷界面中打印模拟电位值  ser = SerialMgt() # 初始化串口对象  ser.write(adc.value()) # 将变量value写入到串口中 |

**6.舵机控制**

代码说明：控制D4引脚上的舵机旋转到150角度

代码范例：

|  |
| --- |
| from xugu import Servo #从xugu库中导入Servo类  servo = Servo(4) # 初始化4号引脚，并连接到舵机  servo.angle(150) # 将舵机旋转150° |

注：舵机转动的时间暂时不能控制。

**7.串口通讯**

代码说明：读取模拟传感器通过串口输出（将Arduino的串口转发到PC的串口）

代码范例：

|  |
| --- |
| # 在虚谷端运行  from xugu import Pin, ADC, SerialMgt #从xugu库中导入Pin类、ADC类、SerialMgt类  p = Pin("A0") # 初始化A0引脚  adc = ADC(p) # 初始化模拟信号对象  value = adc.value() # 读取A0引脚上传感器的电位值  ser = SerialMgt() # 初始化串口对象  ser.write(value) # 将读取到的电位值写入到串口中 |
| # 在PC端运行  import serial # 导入serial类  ser = serial.Serial("COM3") # 初始化COM3口的串口对象（假设PC和虚谷连接的是COM3端口）  ser.baudrate = 115200 # 设置串口速率值，虚谷的默认波特率是115200  if ser.isOpen(): # 如果串口是连接状态  print(ser.read(100) # 读取该串口的值并打印到输出端 |

注：虚谷的串口默认波特率是115200，可以用ser.baudrate 来设置，如ser.baudrate = 115200

**8.串口转Web**

代码说明：**读取传感器并显示在Web页面**

代码范例：

|  |
| --- |
| from xugu import Pin, ADC, Save\_value # 从xugu库导入Pin类、ADC类、save\_value方法  import serial # 导入serial对象  p = Pin("A0") # 初始化A0引脚  adc = ADC(p) # 初始化模拟信号对象  value = adc.value() # 读取A0引脚上传感器的电位值  save\_value("A0", value) # 将变量value保存到数据库中，供serial2web服务调用，关键字是“A0” |

注：虚谷号会运行一个名为serial2web的服务，通过浏览器可以访问板载Arduino的各个引脚的数值。第一个“A0”表示引脚名称，第二个“A0”只是作为数据库的关键字（key值）。

Web访问地址：[http://192.168.1.100:5000/read\_adc/[关键字](http://192.168.1.100:5000/read_adc/%5b关键字)]

例如，获取关键字是“A0”的引脚值，地址为：http://192.168.1.100:5000/read\_adc/a0

**9.读取I2C设备**

代码说明：**读取I2C设备**

代码范例：

|  |
| --- |
| from xugu import I2C # 从xugu库中导入 I2C类  i2c = I2C(I2C.ANALOG, 3, 4) # 初始化i2c设备，3号引脚作为时钟线，4号引脚作为数字线  data = i2c.read(0x42, 0x10) # 从0x42地址的0x10寄存器读取数据 |

**10.写入I2C设备**

代码说明：**写入I2C设备**

代码范例：

|  |
| --- |
| from xugu import I2C # 从xugu库中导入 I2C类  i2c = I2C(I2C.ANALOG, 3, 4) # 初始化i2c设备，3号引脚作为时钟线，4号引脚作为数字线  data = i2c.write(‘xy’, 0x42, 0x10) # 向0x42这个地址的的0x10寄存器上写数据 |