**xugu库的使用**

xugu库中有Pin、Servo、I2C、SerialMgt、LED、XuguLog等类。

## 控制I/O引脚：Pin类

Pin类用于控制I/O引脚，具有设置引脚模式（IN，OUT）的属性和读写电平状态的方法。

1. **构建**

**Pin(pin\_num, pin\_model）**

pin\_num传入引脚标号，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。

pin\_model 为引脚模式，Pin.IN是输入模式，Pin.OUT是输出模式。

1. **方法**

**Pin.read\_digital ()**

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。

**Pin.read\_analog ()**

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。

**Pin.write\_digital (value)**

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。

**Pin.write\_analog (value)**

给引脚设置模拟值。value指要设置的模拟值，数据范围在0和1023之间。该方法在输出模式有效。

## 舵机控制：Servo类

该类用于控制舵机转到指定角度。

1. **构建**

**Servo(pin\_num）**

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。

1. **方法**

**Servo. write\_angle (value)**

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。

## 读写I2C设备：I2C类

该类用于读写I2C从设备。

注：该类不能用于读写虚谷号主芯片的I2C总线。

1. **构建**

**I2C(time=0)**

time 指I2C总线连续读写的间隔时间，单位是毫秒（ms），默认值是0。

1. **方法**

**I2C .readfrom(address, register, read\_byte)**

读取I2C设备。address为I2C从设备的地址，register为从设备的寄存器，read\_byte为一次读取的字节数量。

**I2C .writeto(address, args)**

向I2C设备中写入内容。address为I2C从设备的地址，args是要发送到设备的可变字节数，作为列表传入。

## 串口对象：SerialMgt类

该类用于虚谷号和PC之间的串口通信。

1. **构建**

**SerialMgt(port, baudrate)**

port指虚谷号连接pc的串口号，baudrate为串口波特率；

注：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。

1. **方法**

**SerialMgt.read(bytes)**

从串口中读取数据，bytes为读取的字节数，默认为100。

**SerialMgt.write(data)**

向串口中写入数据，data为写入的数据，类型为String。

## LED类

该类用于LED的简易控制。

1. **构建**

**LED(pin\_num)**

Pin\_num为数字引脚编号，范围0~19。

**注：虚谷号已经在第13号引脚内置了LED。**

1. **方法**

**high()**

给引脚一个高电位，只有在输入模式有效，当该引脚接入LED灯的时候，灯会点亮。

**low()**

给引脚一个低电位，只有在输入模式有效，当该引脚接入LED灯的时候，灯会熄灭。

**on()**

等价与high()。

**off()**

等价与low()。

## XuguLog: 日志输出类

该类用于日志输出。程序运行过程中，会将日志信息追加到日志文件中。

1. **构建**

**XuguLog(filename)**

初始化该类的时候，会自动生成一个名为filename的日志文件，后缀为.log，生成的文件与运行的python程序在同一个目录下。

1. **方法**

**XuguLog.write(value)**

将日志信息写入到日志文件中，value为要写入的内容，类型为String。