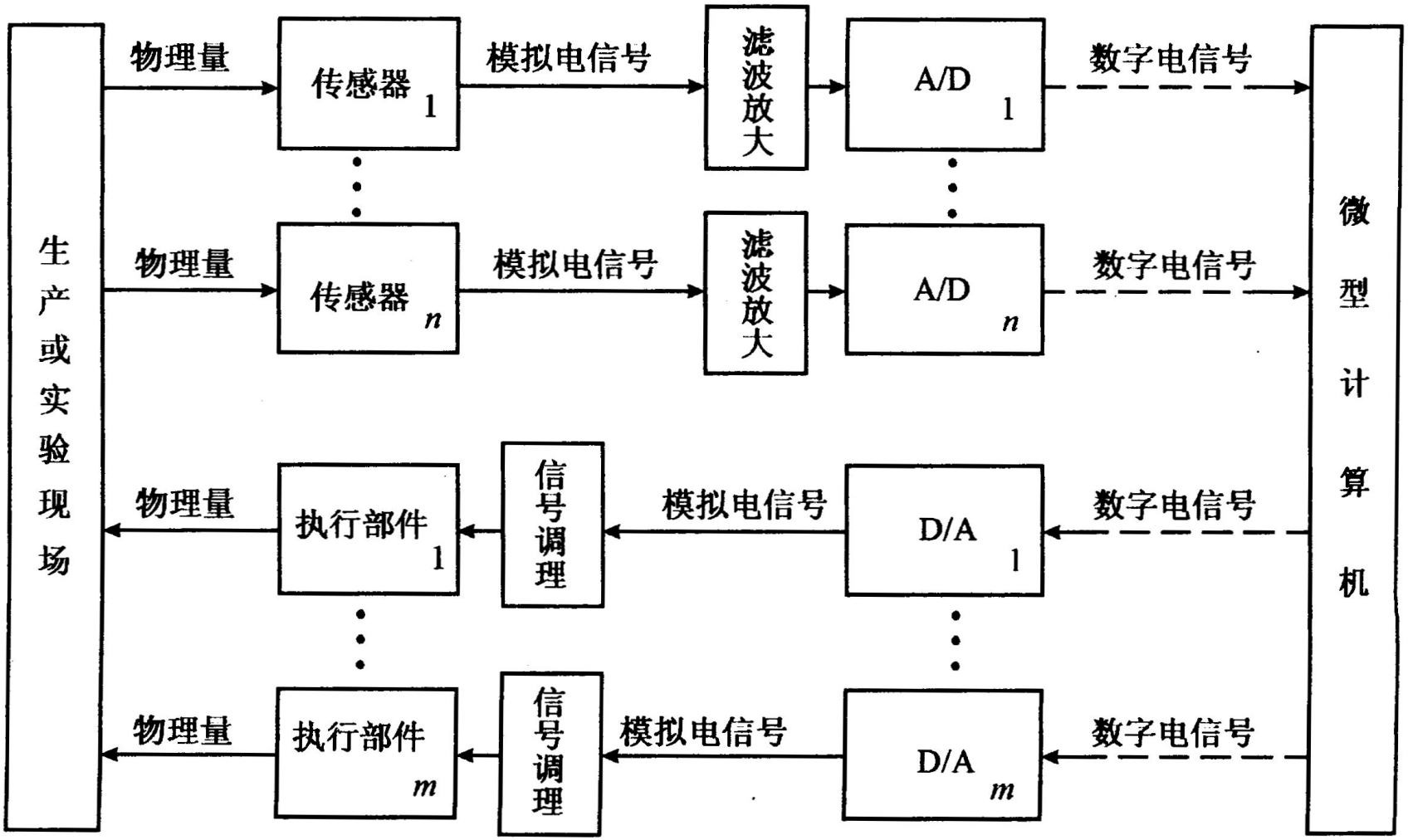
模数转换

电子系统导论教学团队

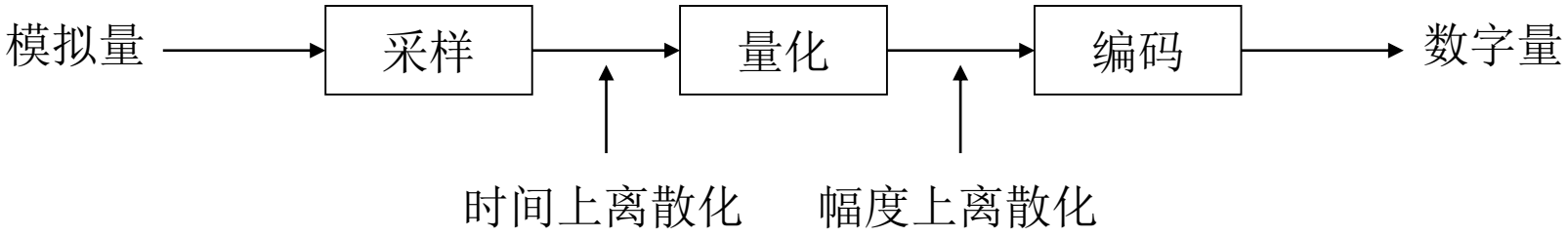
实验目的

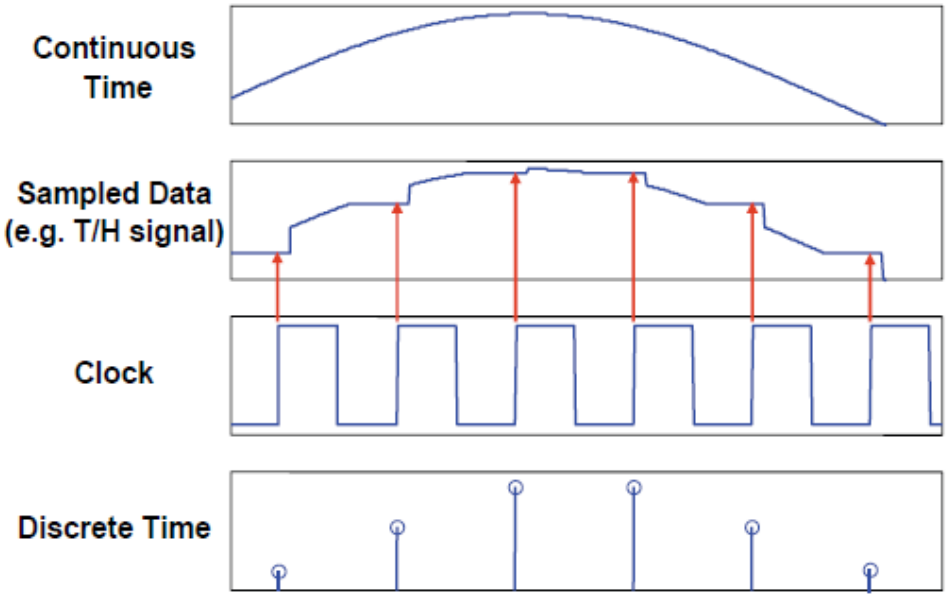
- 了解模数转换的基本概念
- 初步了解信号发生器、示波器的使用
- 以ADC0832为例，掌握ADC的基本使用方法

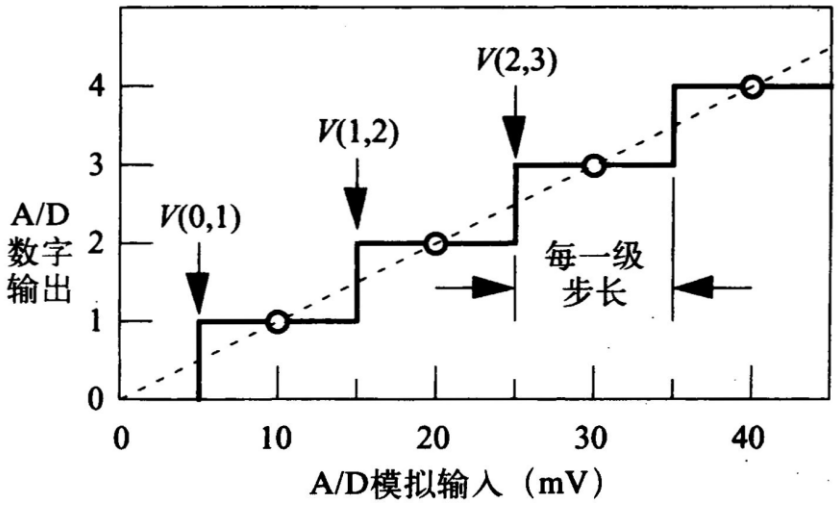
模拟量与数字量


生产或实验现场
物理量
传感器 1
传感器 n
模拟电信号
滤波放大
滤波放大
A/D 1
A/D n
数字电信号
数字电信号
微型计算机
物理量
执行部件 1
执行部件 m
信号调理
信号调理
模拟电信号
模拟电信号
D/A 1
D/A m
数字电信号
数字电信号

模数转换器ADC


模拟量
采样
量化
编码
数字量
时间上离散化
幅度上离散化

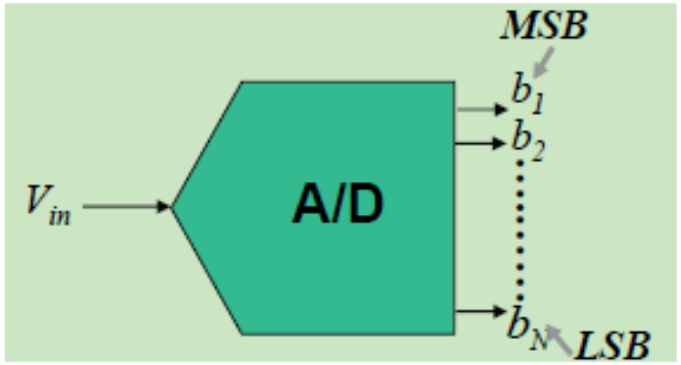

Continuous Time
Sampled Data (e.g. T/H signal)
Clock
Discrete Time


A/D 数字输出
V(0,1)
V(1,2)
V(2,3)
每一级步长
0
1
2
3
4
0
10
20
30
40
A/D模拟输入（mV）

模数转换器ADC

- 将模拟电压量转换成离散数值，通常用二进制或16进制表示

$$D = Vin/\Delta$$



$N = \#\ of\ bits$

$V_{FS} = full\ scale\ output$

$\Delta = min.\ resolvable\ input \rightarrow 1LSB$

$$\Delta = \frac{V_{FS}}{2^N}$$

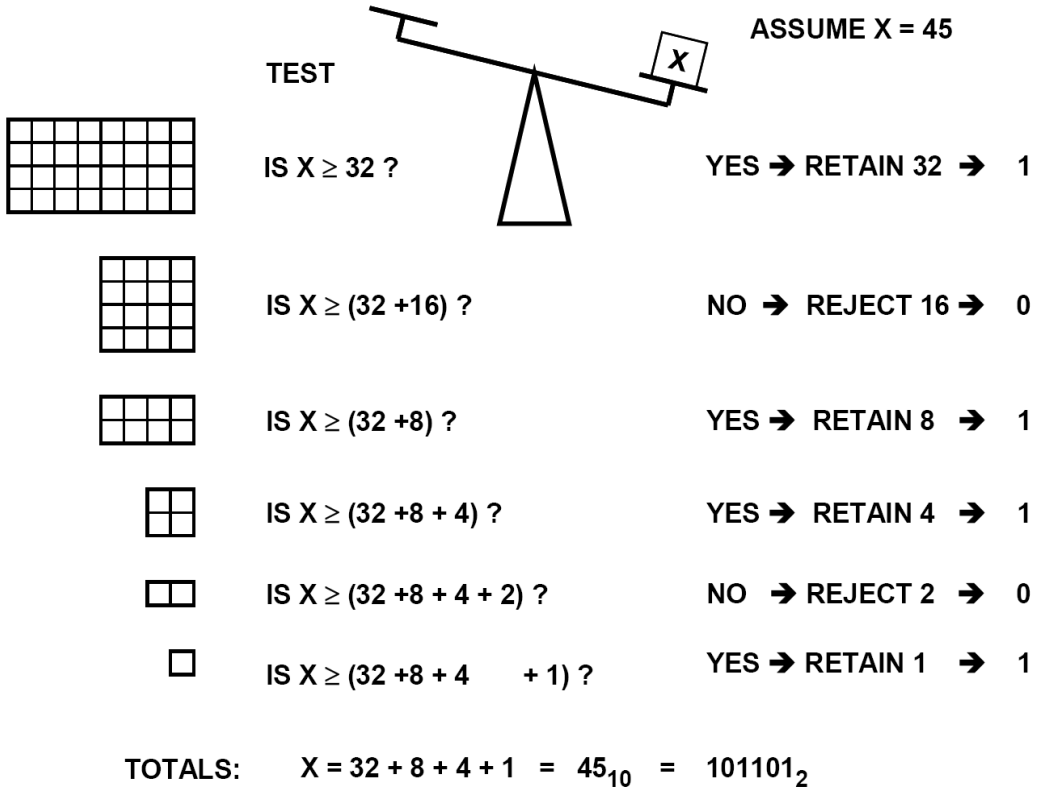
$$or\ N = log_2 \frac{V_{FS}}{\Delta} \rightarrow resolution$$

ADC0832简介

- ADC0832

  ADC0832 是美国国家半导体公司生产的一种8位分辨率、双通道A/D转换芯片。其工作原理为逐次逼近型。

✓ 以称重为例，概述逐次逼近ADC的基本工作原理



ADC0832简介

- ADC0832

  ADC0832 是美国国家半导体公司生产的一种8位分辨率、双通道A/D转换芯片。其工作原理为逐次逼近型。

- 特点
  - 输入输出电平与TTL/CMOS相兼容；
  - 5V电源供电时输入电压在0~5V之间；
  - 该ADC的满量程参考电压直接设置为电源电压；

  Note：本实验中用GPIO3.3V供电，**输入电压必须在0~3.3V之间；**

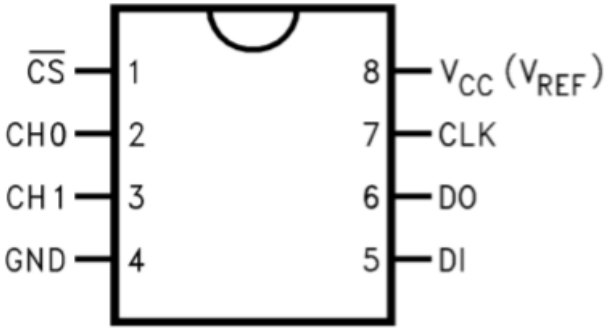
  - 工作频率为250KHz，转换时间为32μS；
  - 一般功耗为15mW；
  - 8P、14P—DIP（双列直插）、PICC 多种封装；
  - 商用级芯片温宽为0℃ to +70℃ ，工业级芯片温宽为-40℃ to +125℃；

ADC0832简介

- 芯片接口说明
  - -CS 片选使能，低电平芯片使能。
  - CH0 模拟输入通道0，或作为IN+/-使用。
  - CH1 模拟输入通道1，或作为IN+/-使用。
  - GND 芯片参考0 电位（地）。
  - DI 数据信号输入，选择通道控制。
  - DO 数据信号输出，转换数据输出。
  - CLK 芯片时钟输入。
  - Vcc/REF 电源输入及参考电压输入（复用）。
- 参考电压：用作模拟信号输入电压的最大参考值。若输入参考电压+3V，则认为+3V为模拟信号的最高电压，将0～ +3V等分为256级，每一级对应0～255的一个数字信号从DO输出。如：输入的模拟信号大小为+1.5V ，则输出127。

Top View

ADC0832 2-Channel MUX
Dual-In-Line Package (N)



00558331

COM internally connected to GND.

V_{REF} internally connected to V_{CC}.

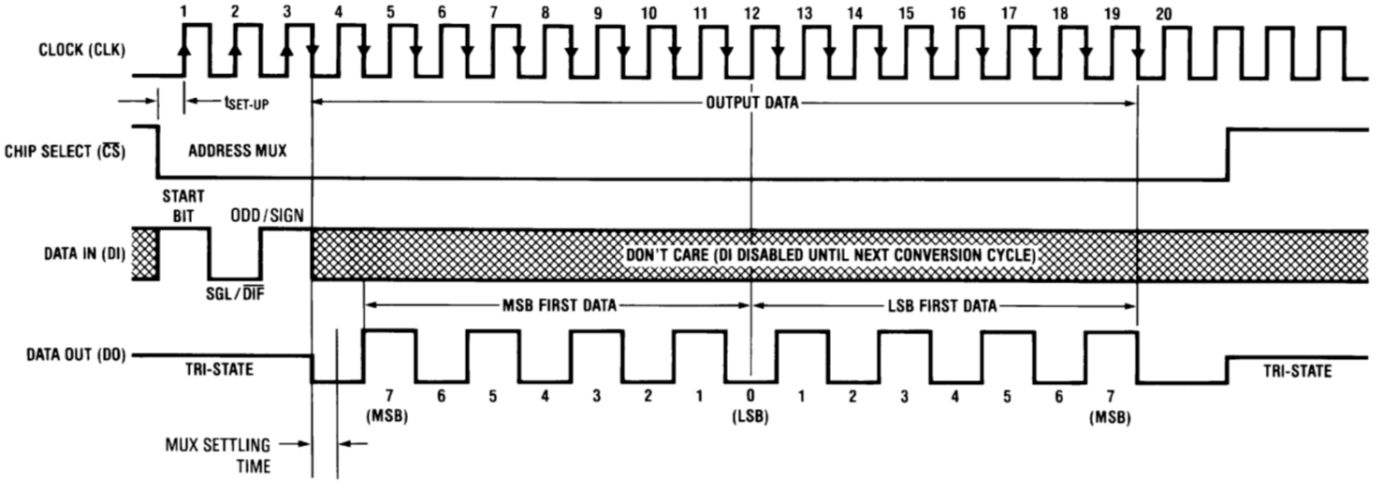
Top View

ADC0832简介

- 应用接口与工作时序：正常情况下ADC0832与单片机的接口应为4条数据线，分别是-CS、CLK、DO、DI。但由于D0端与DI端在通信时并未同时有效并与树莓派的接口是双向的，所以电路设计时可以将D0和DI并联在一根数据线上使用。当-CS输入端高电平时芯片禁用。当要进行A/D转换时，须先将-CS置于低电平并且保持低电平直到转换完全结束。
- 在第1个时钟脉冲的下降沿之前DI端必须是高电平，表示启始信号。在第2、3个脉冲下降沿之前DI端应输入2位数据用于选择模拟信号通道。

Timing Diagrams (Continued)

ADC0832 Timing



ADC0832简介

- 工作原理：如表1所示，当这两位数据为“1”、“0”时，设置CH0为单端输入；
- 当2位数据为“1”、“1”时，设置CH1为单端输入；
- 当2位数据为“0”、“0”时，将CH0作为正输入端IN+，CH1作为负输入端IN-进行差分输入；
- 当2位数据为“0”、“1”时，将CH0作为负输入端IN-，CH1作为正输入端IN+进行差分输入。
- 后续示例和代码中设置CH0单端输入。

TABLE 6. MUX Addressing: ADC0832 Single-Ended MUX Mode

MUX Address		Channel #	
SGL/$\overline{\text{DIF}}$	ODD/SIGN	0	1
1	0	+	
1	1		+

COM is internally tied to A GND

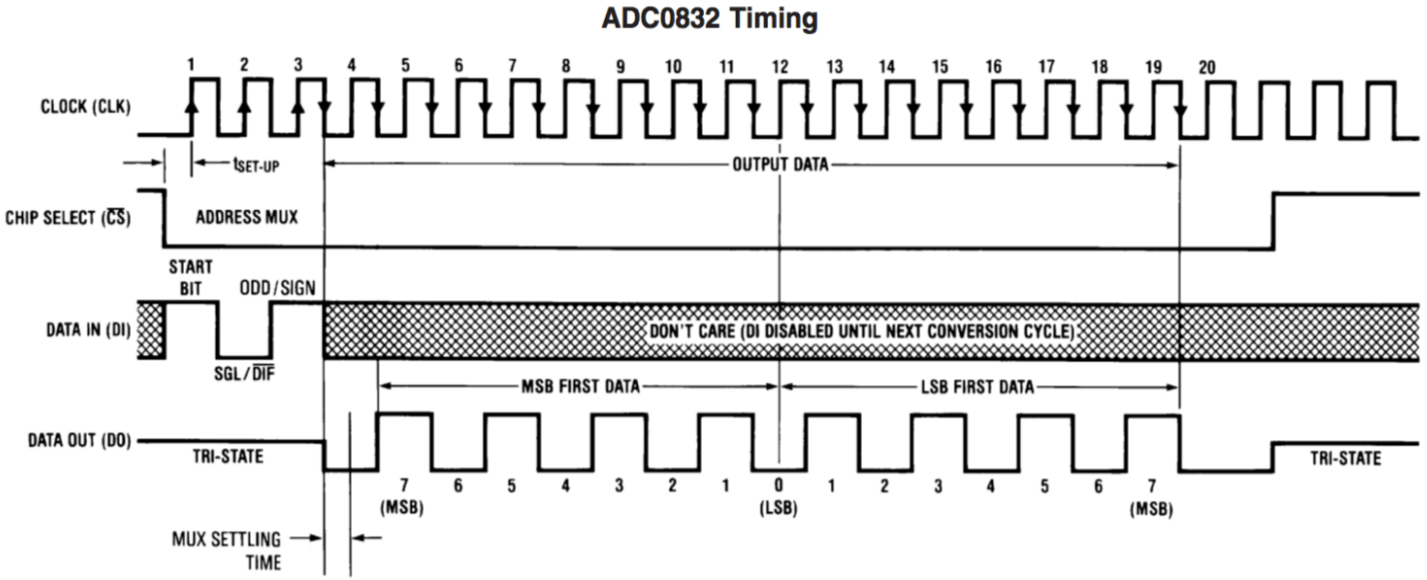
TABLE 7. MUX Addressing: ADC0832 Differential MUX Mode

MUX Address		Channel #	
SGL/$\overline{\text{DIF}}$	ODD/SIGN	0	1
0	0	+	−
0	1	−	+

ADC0832简介

- 工作原理：到第3个脉冲的下降沿之后输出D0进行转换数据的读取。从第4个脉冲下降沿开始由DO端输出转换数据最高位DATA7，随后在每一个脉冲的下降沿D0端输出下一位数据，直到第11个脉冲时发出最低位数据DATA0。
- 紧接着，从此位开始输出一个相反顺序的数据，即从第11个字节的下降沿输出DATD0，随后输出8位数据，到第19个脉冲时数据输出完成。

Timing Diagrams (Continued)



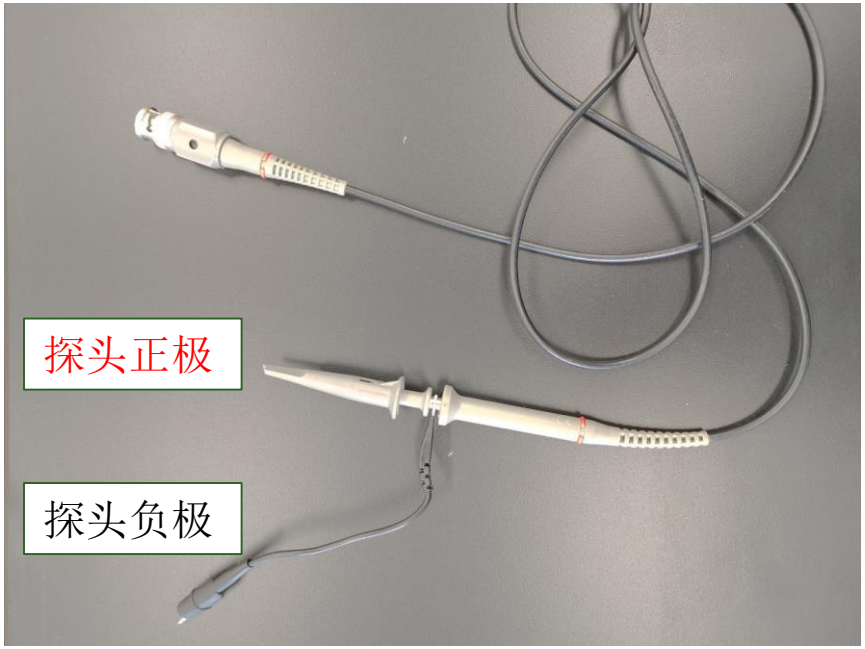
实验1：树莓派控制ADC0832测量电位器电压

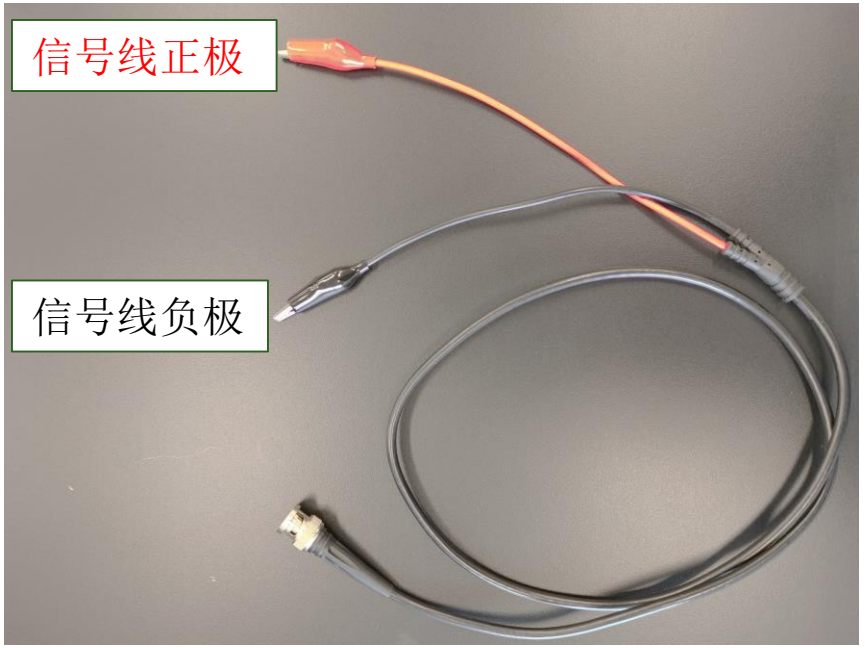
- 实验目的：
  - 以ADC0832为例，初步掌握ADC的使用方法

实验准备-库安装

- 安装Matplotlib与NumPy（若已安装可跳过）
  - Matplotlib 是一个 Python 的 2D绘图库
  - 在终端中安装Matplotlib：
  - $ sudo apt install python3-matplotlib
  - NumPy系统是Python的一种开源的数值计算扩展。
  - 在终端中安装NumPy：
  - $ sudo apt install python3-numpy
- 安装wiringpi（若已安装可跳过）
  - $ pip3 install wiringpi
- 将ADC0832.py拷贝至树莓派，文件可在课程ftp中下载。

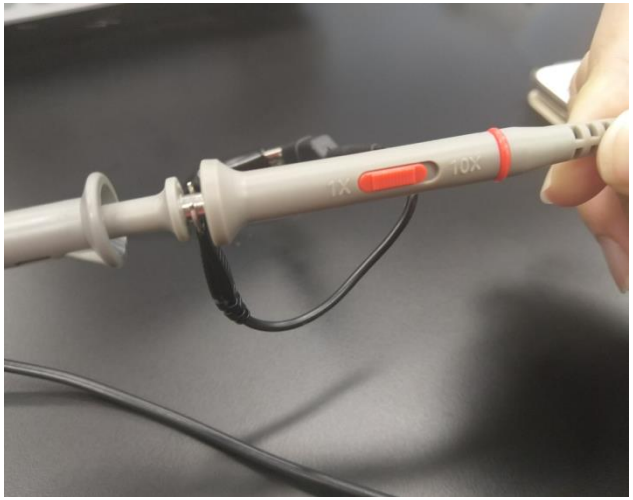
示波器探头与信号线



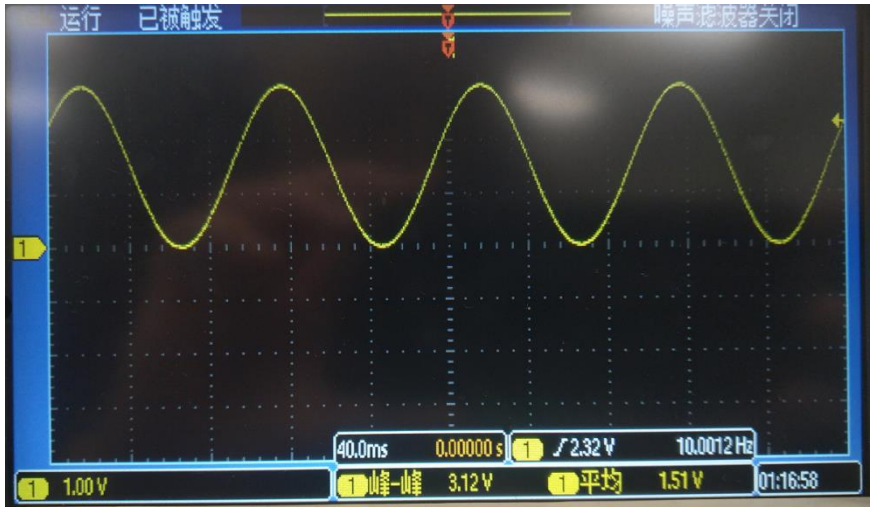


仪器使用

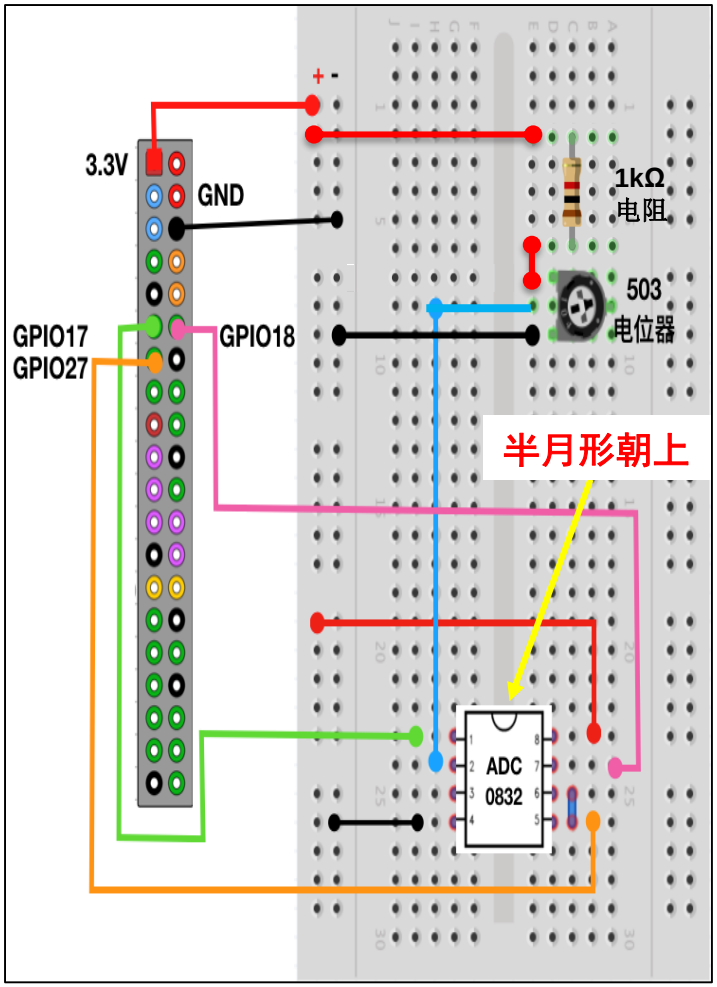
- 连接并配置任意波形信号发生器。
- 用示波器采样，根据探头实际参数设置，调节信号发生器电压值与直流偏置，使电压值在0~3.3V之间，示例中取频率为10Hz。
- 注意示波器采样时要在非自动模式下，手动设置直流耦合。

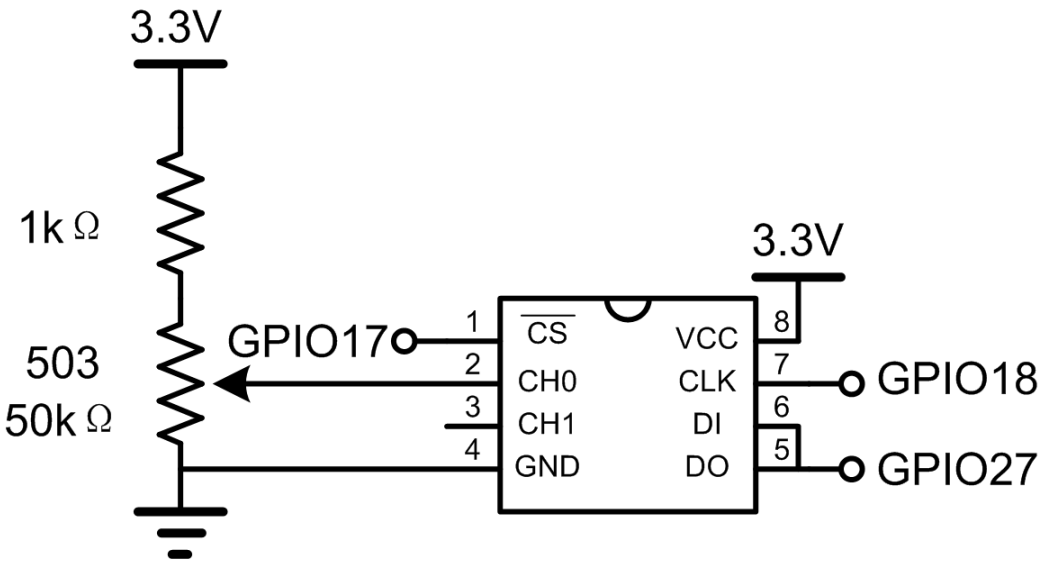


确认衰减开关为1x 还是 10x



电路连接





- 示例中使用503电位器，也可用滑动变阻器代替。
- 电位器一端连接到3.3V，另一端接地，这就允许电位器分压输出0到3.3V之间的电压。
- 同时，探头正极连电位器输出，探头负极连GND（共地）

树莓派控制ADC0832测量电位器电压

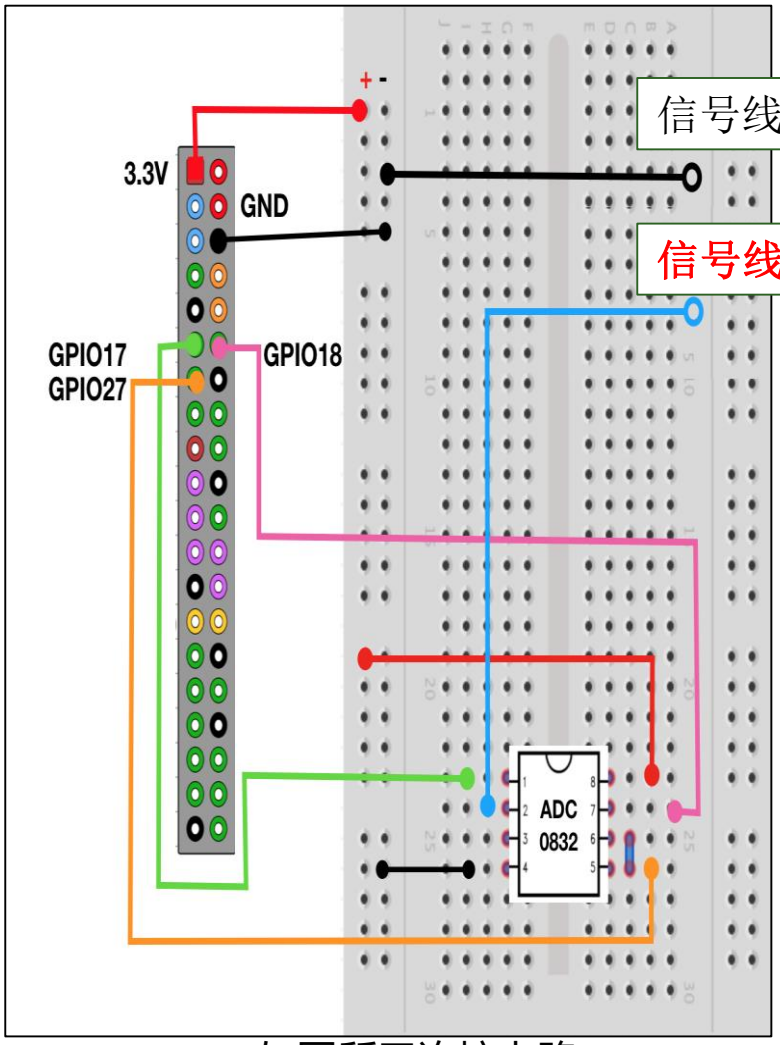
```
import ADC0832
import time
def init():
    ADC0832.setup()
    #def setup(cs=11,clk=12,dio=13):
    #    global ADC_CS, ADC_CLK, ADC_DIO
    #    ADC_CS=cs
    #    ADC_CLK=clk
    #    ADC_DIO=dio
    #    GPIO.setwarnings(False)
    #    GPIO.setmode(GPIO.BOARD)
    #    GPIO.setup(ADC_CS, GPIO.OUT)
    #    GPIO.setup(ADC_CLK, GPIO.OUT)
def loop():
    while True:
        digitalVal=ADC0832.getResult()
        # Get ADC result, input channal
        # getResult()函数代码实现ADC0832工作原理
        # 得到0~255之间的一个数
        # 详细参见ADC0832.py
        print(3.3*float(digitalVal)/255)
        # 转换为电压量
        time.sleep(0.2)
if __name__=='__main__':
    init()
    try:
        loop()
    except KeyboardInterrupt:
        ADC0832.destroy()#调用GPIO.cleanup
        print("The end!")
```

- 在终端中运行，改变电位器电阻值，即可测得相应的电压。

实验2：树莓派控制ADC0832测量正弦电压

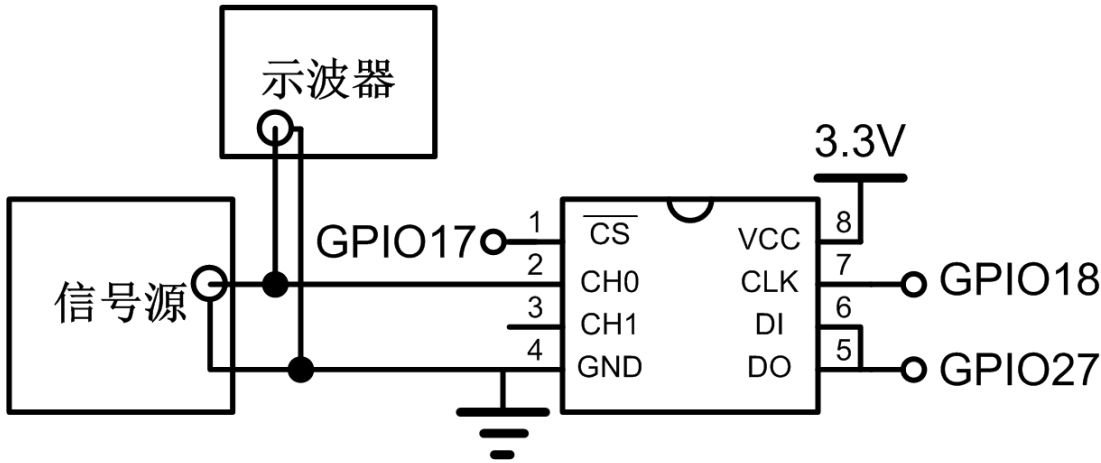
- 实验目的：
  - 初步了解信号源与示波器的使用
  - 以ADC0832为例，掌握ADC的使用方法

电路连接



信号线负极与探头负极共地

信号线正极 并联 探头正极



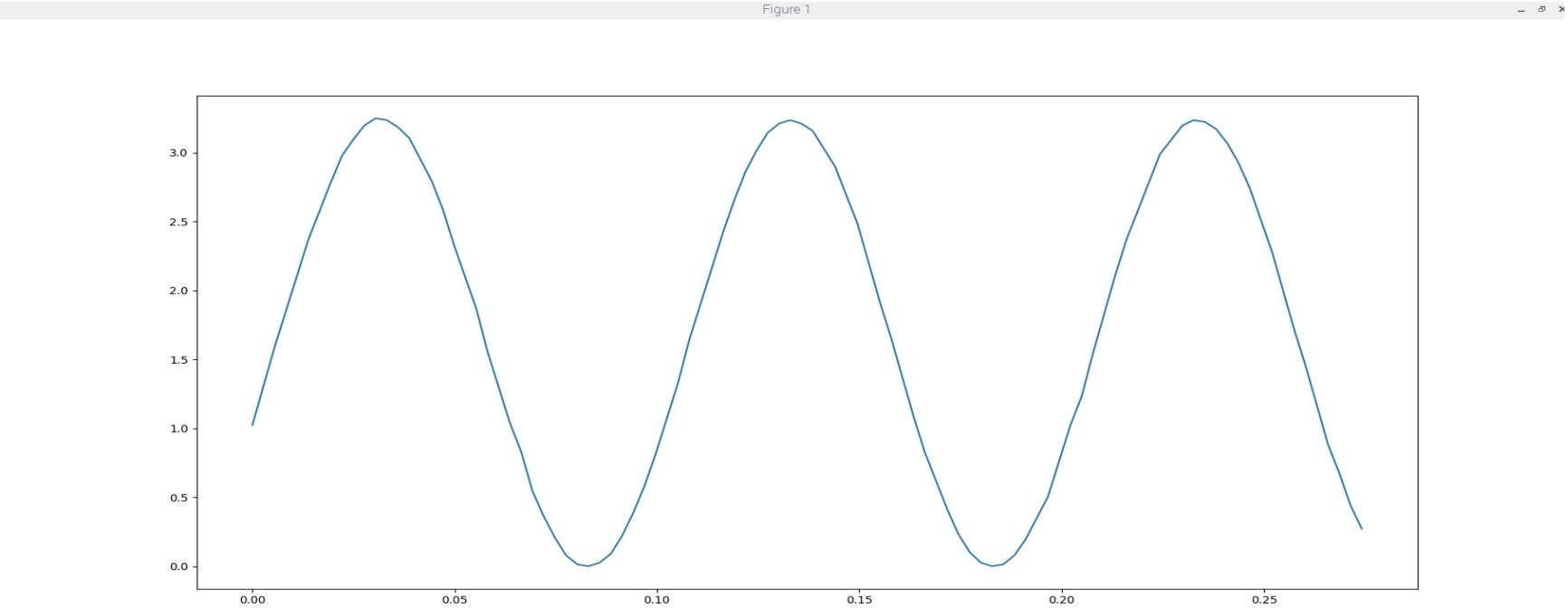
- 如图所示连接电路

树莓派控制ADC0832测量正弦电压

```
import ADC0832
import time
import numpy as np
import matplotlib.pyplot as plt
def init():
    ADC0832.setup()
def loop():
    n=0
    t=0
    y=[]
    x=[]
    t_org = time.perf_counter()
    while n<1000:
        digitalVal=ADC0832.getResult()
        y.append(3.3*float(digitalVal)/255)
        x.append(time.perf_counter() - t_org)
        # or time.process_time()
        n=n+1
    # t=time.perf_counter()-t
    # or time.process_time()
    plt.plot(x,y)
    plt.show()
if __name__=='__main__':
    init()
    loop()
    ADC0832.destroy()
    print("The End")
```

- 可根据需要在循环中加入time.sleep（）函数或调整取点个数
- 注意每次运行程序后要及时关闭图像窗口
- 注意：sleep函数只有毫秒级精度（Linux OS进程调度所致），因此无法保证精确定时采样。可用wiringpi中的delayMicroSeconds函数提高精度(10us级)

树莓派控制ADC0832测量正弦电压



- 在终端中运行程序，即可得到正弦波曲线图

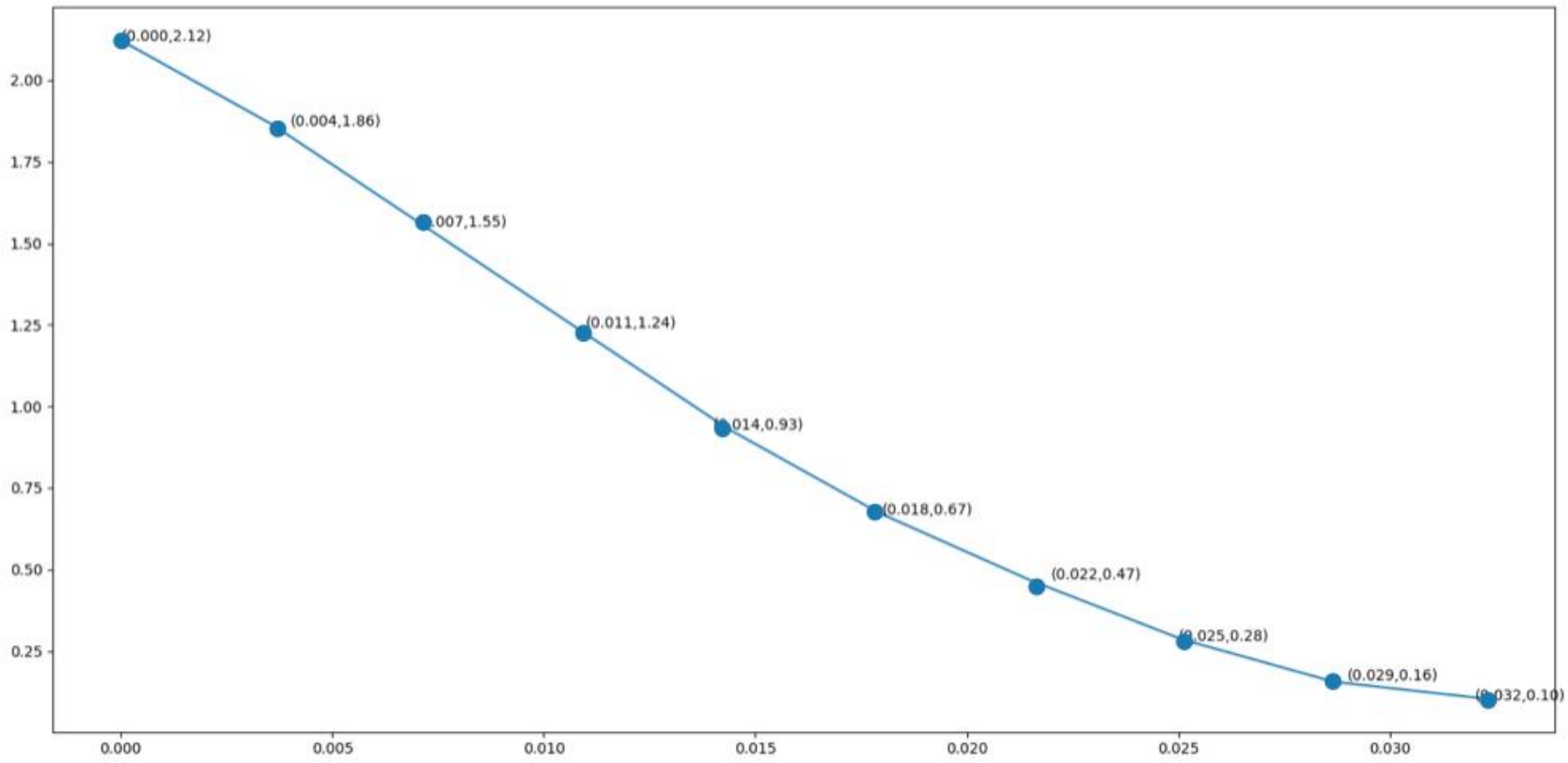
基于matplotlib的简易图像处理

```
import ADC0832
import time
import numpy as np
import matplotlib.pyplot as plt
def init():
    ADC0832.setup()
def loop():
    n=0
    i=0
    y=[]
    x=[]
    t=time.process_time()

    while n<10:
        digitalVal=ADC0832.getResult()
        y.append(3.3*float(digitalVal)/255)
        x.append(time.process_time()-t)
        n=n+1
    #plt.axis([0.1,0.2,0.4,0.8])
    #前后两组参数分别表示x、y轴范围
    #等价于plt.xlim(0.1,0.2) plt.ylim(0.4,0.8)
    plt.plot(x,y,'-o')
    while i<10:
        x1="%.3f"%x[i]
        y1="%.2f"%y[i]
        text='{'+str(x1)+','+str(y1)+'}'
        plt.text(x[i],y[i],text)
        i=i+1
    plt.show()
if __name__=='__main__':
    init()
    loop()
    ADC0832.destroy()
    print("The End")
```

- 控制坐标轴范围：通过plt.axis()、plt.ylim()、plt.xlim()函数可以控制坐标轴的范围从而局部放大函数图象；
- 显示坐标点：plt.plot(x,y,’o’)可显示数据散点，plt.plot(x,y,’-o’)可将获得带数据点的折线图（也可将‘o’换为‘*’等）；
- 注释点的坐标：通过plt.text()函数为指定位置加上文本注释，示例中为十个点添加注释；
- 更多的图像处理方法参考matplotlib.org

基于matplotlib的简易图像处理



- 在终端中运行程序，得到带数据点的图像

实验内容

- 完成对电位器电压的采集
  - 设置10组电位器的不同分压位置，用ADC完成模数转换，得到相应的模拟电压，并与示波器直接测得的电位器分压输出电压进行比较，计算误差。
- 完成对正弦电压的采集
  - 设置信号发生器输出0-3.3V之间的正弦信号，进行定时采样，并描绘出波形。
- 完成每个实验后需要由教师或者助教验收。

实验报告中需要回答的问题

1. 示波器探头的直流耦合和交流耦合有什么区别？
2. 为什么信号发生器的输出要加直流偏置？

致谢

- 本课件由以下同学协助编写、修改
  - 王建宸（16307130029）
  - 叶诏辉（16307130039）
  - 江逸舟（18110720013）