

实验4 传感器实验（下）

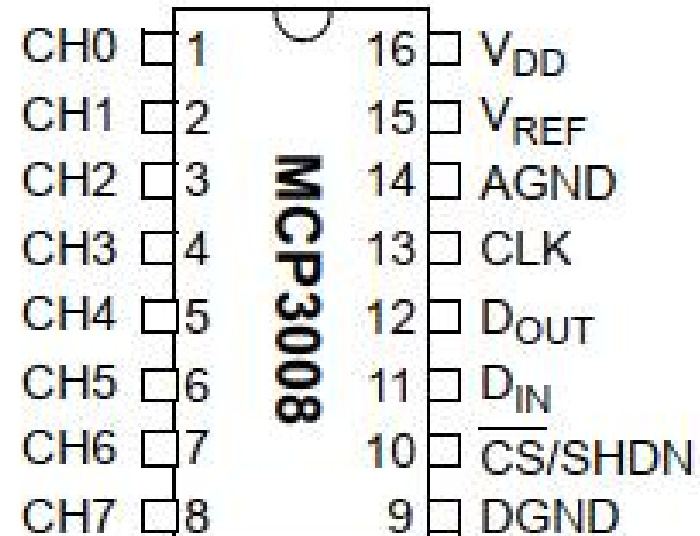
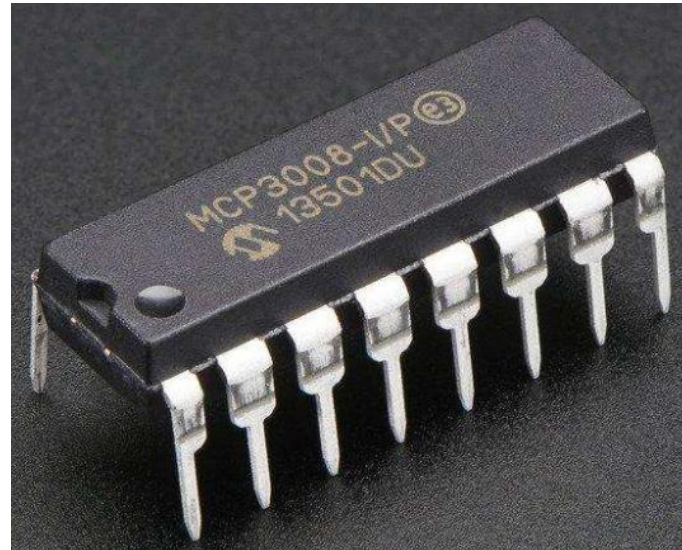
2024. 3. 21

传感器

- 能感受规定的被测量并按照一定的规律(数学函数法则)转换成可用信号的器件或装置,通常由敏感元件和转换元件组成
- 传感器本质是一个模拟信号转数字信号的数据采集器ADC
(Analog-Digital Converter)
- 所以, 通用ADC器件+模拟信号传感器同样可以完成数据传感采集
 - 如MCP3008+TMP36
- DHT11数字式温湿度传感器
- 其他接口的传感器
 - 如I2C接口, ADXL345加速度计

MCP3008--ADC

- 8通道
- 10bit($2^{10}=1024$)
- 工作电压：2.7V—5.5V
- SPI口
- 封装：PDIP、TSSOP等



SPI--Serial Peripheral Interface

- 串行外设接口（Serial Peripheral Interface）的缩写，是一种高速，全双工，同步的通信总线，并且在芯片的管脚上只占用四根线

- 工作原理：

以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，需要至少4根线，事实上3根也可以（单向传输时）。分别是MISO（主设备数据输入）、MOSI（主设备数据输出）、SCLK（时钟）、CS（片选）。

1. MISO– Master Input Slave Output，主设备数据输入，从设备数据输出；
2. MOSI– Master Output Slave Input，主设备数据输出，从设备数据输入；
3. SCLK – Serial Clock，时钟信号，由主设备产生；
4. CS – Chip Select，从设备使能信号，由主设备控制。

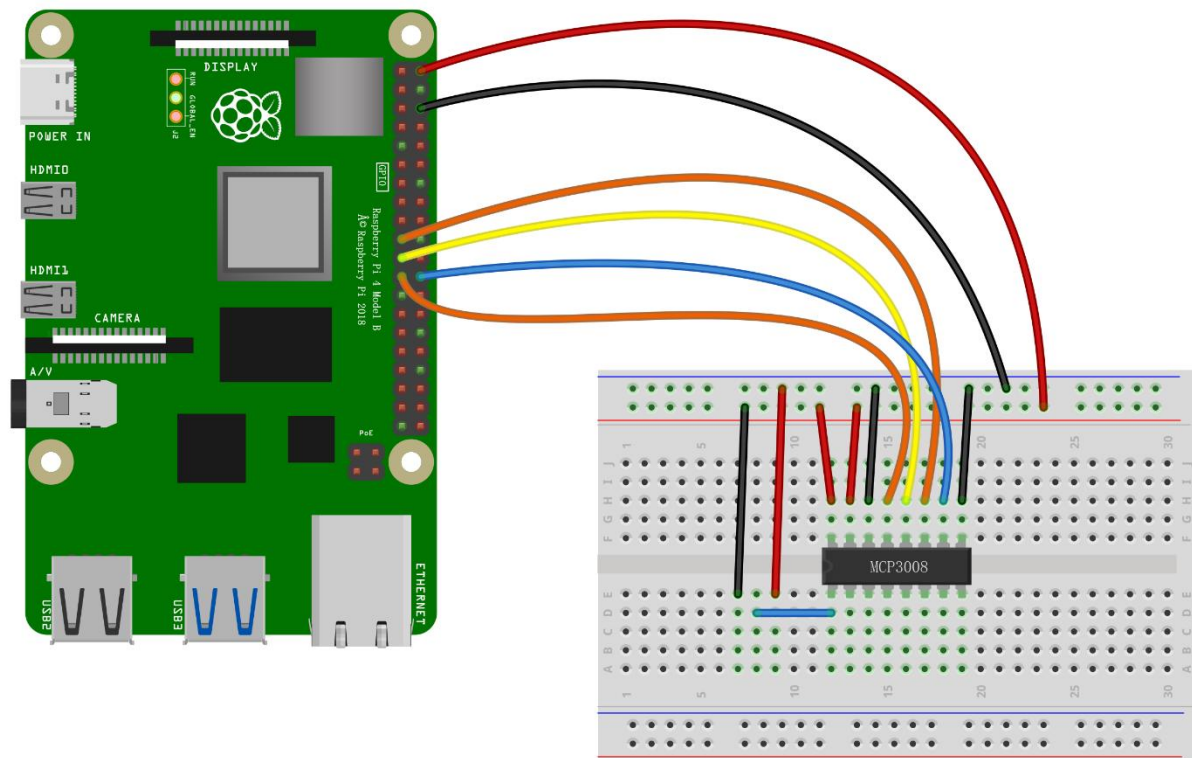
树莓派的SPI



Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21

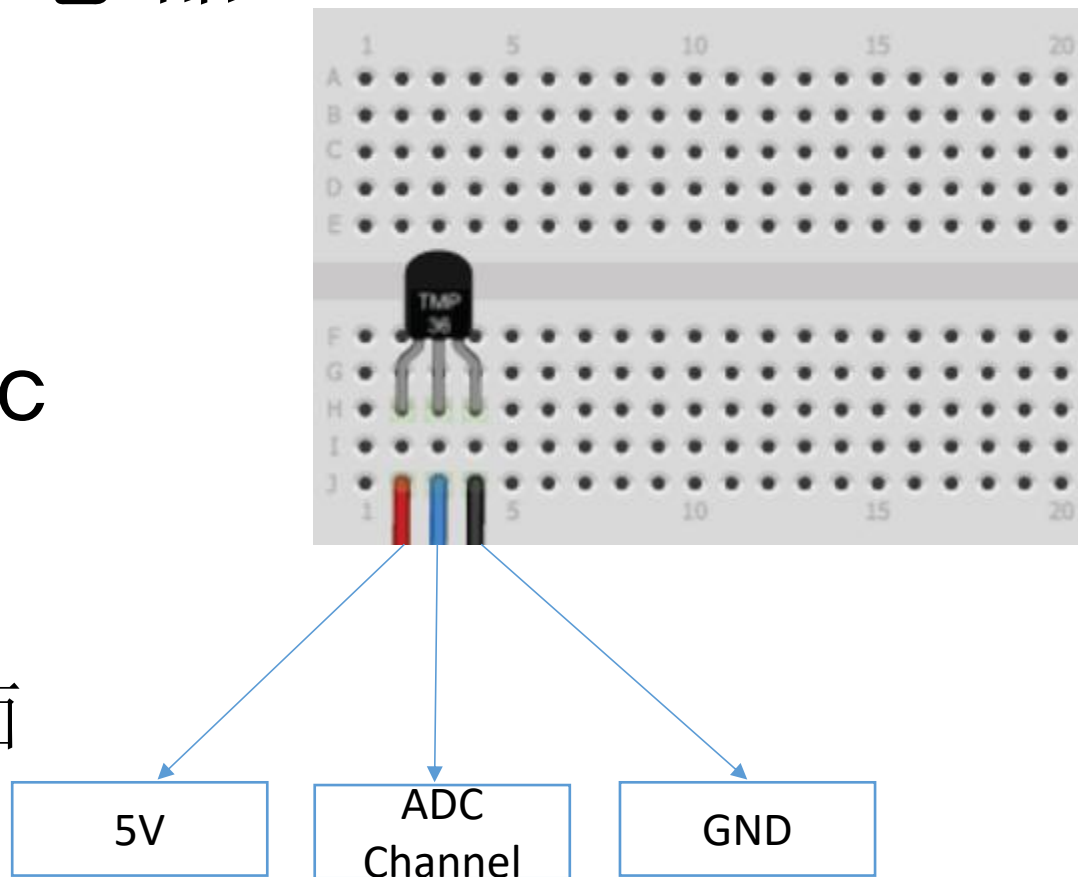
SPI连线

Raspberry Pi	MCP3008
5V	VDD
5V	Vref
GND	AGND
SPI0 SCLK	CLK
SPI0 MISO	Dout
SPI0 MOSI	Din
SPI0 CS0	CS
GND	DGND



TMP36—温度传感器

- 工作电压：2.7V至5.5V
- 摄氏温度传感器
- 测温范围：-40°C~150°C
- 连接ADC接口简单
- 封装：TO92
- 器件有标记的是扁平面



注意：电源和地不能接反，会烧器件

```
from gpiozero import MCP3008

pot = MCP3008(channel=0)

while True:
    print(pot.value)
```

注意：value不是tmp36的
输出电压值Vout，而是
 V_{out}/V_{ref}

```
from gpiozero import MCP3008
from time import sleep

def convert_temp(gen):
    for value in gen:
        yield (value * 3.3 - 0.5) * 100

adc = MCP3008(channel=0)

for temp in convert_temp(adc.values):
    print('The temperature is', temp, 'oC')
    sleep(1)
```


DHT11连线

Raspberry Pi	DHT11
5V	VDD
GND	GND
GPIO	DOUT



DHT11

```
import Adafruit_DHT,time

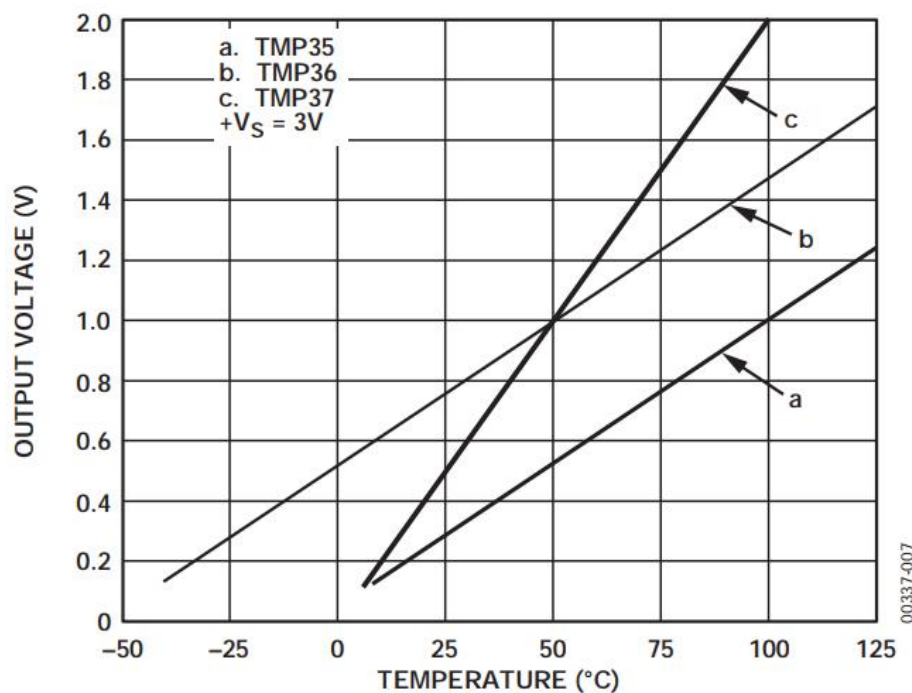
sensor=Adafruit_DHT.DHT11

gpio=19

while True:
    hum, tem = Adafruit_DHT.read_retry(sensor, gpio)
    if hum is not None and tem is not None:
        print('Temp={0:0.1f}*C Humi={1:0.1f}%'.format(tem, hum))
    else:
        print('Failed ')
    time.sleep(1)
```

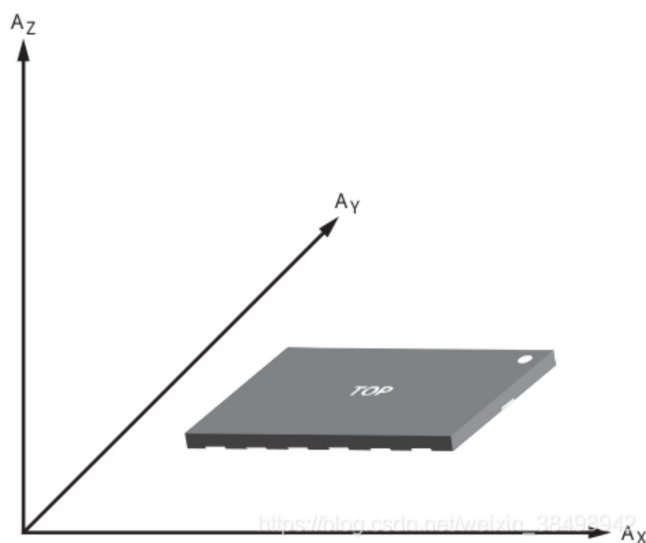
作业1

- 温度检测：测量环境温度，单位摄氏度
- 注意分析ADC输入电压值转化成温度值的计算，并写入报告
- 报告、代码、演示视频打包提交至课程平台，截止：2024年3月28日



加速度计传感器

- 芯片：ADXL345，3轴加速度传感器，可测量X、Y、Z三个方向上的加速度，接口I2C



- 运动检测：当 ADXL345 沿检测轴正向加速时，它对正加速度进行检测
- 重力检测：在检测重力时，当检测轴的方向与重力的方向相反时检测到的是正加速度

常见应用场景

- 汽车安全
- 游戏控制
- 自动图像翻转
- 防抖功能
- 电子罗盘倾斜校正
- 计步器功能
- **GPS**导航系统的盲区补偿
- 行人跌倒检测

参数

- Working voltage: 3.0 - 5.5V
- Test Range: ± 16
- Sensitivity: 3.9mg / LSB
- Standby Current: 0.1 μ A (Under stand mode Vcc = 2.5 V (typical))
- 10000 g high shock survivability
- ECOPACK[®]RoHS and “Green” compliant
- Suli-compatible Library



I2C

- Inter-Integrated Circuit，有时也写为IIC
- 数据线SDA和时钟线SCL
- 不论总线上挂多少个设备，只需要两根信号线（时钟SCL和数据SDA）
- 两根信号线都通过合适阻值的上拉电阻连接到正电源
- 每个从设备都有一个7位的地址，可供寻址用。主设备必须知道这些从设备的地址以便同指定的一个从设备进行通信
- 所有传输均由“主设备”发起和终止;“主设备”可以将数据写入一个或多个“从设备”或从“从设备”请求数据
- 在系统中“主”和“从”不是固定的，任何一个设备都可以作为“主”或“从”

连接方法

Raspberry Pi	ADXL345
3.3V或5V	VCC
GND	GND
SDA	SDA
SCL	SCL

SMBus

- SMBus基于I2C协议，以I2C为物理层，因此采用两线接口和设备寻址进行通信

- 导入方法一：

```
import smbus
```

- 导入方法二：

```
from smbus import SMBus
```

- 创建smbus对象时，需要指定I2C总线的编号：

```
bus = smbus.SMBus(1)
```

- 读取数据
 - address = 0x53 #设备地址
 - register = 0x32 #寄存器地址
 - data = bus.read_byte_data(address, register)
 - data = bus.read_block_data (address, register)
-
- 写入数据
 - data = 0x01 #要写入的数据
 - bus.write_byte_data(address, register, data)

https://blog.csdn.net/Jacob_zhj/article/details/90518363

地址	寄存器名称	类型	默认值	说明	设置值	设置的功能
0	DEVID	只读	0xE5	器件ID	只读	
1-1C	Reserved			保留, 不要操作	保留	
1D	THRESH_TAP	读/写	0x00	Tap的门限	不使用	
1E	OFSX	读/写	0x00	X轴失调	0x06	补偿X轴失调, 通过初始化校正获得
1F	OFSY	读/写	0x00	Y轴失调	0xF9	补偿Y轴失调, 通过初始化校正获得
20	OFSZ	读/写	0x00	Z轴失调	0xFC	补偿Z轴失调, 通过初始化校正获得
21	DUR	读/写	0x00	Tap的持续时间	不使用	
22	LATENT	读/写	000	Tap的延迟时间	不使用	
23	WINDOW	读/写	0x00	Tap的时间窗	不使用	
24	THRESH_ACT	读/写	0x00	Activity 的门限	0x20/0x08	设置Activity 的门限为2 g或0.5 g
25	THRESH_INACT	读/写	0x00	Inactivity 的门限	0x03	设置Inactivity 的门限为0.1875 g
26	TIME_INACT	读/写	0x00	Inactivity 的时间	0x02/0x0A	设置Inactivity 的时间为2 s 或 10 s
27	ACT_IACT_CTL	读/写	0x00	Activity/Inactivity使能控制	0x7F/0xFF	使能 X、Y、Z三轴的 Activity和Inactivity功能, 其中Inactivity为AC coupled模式, Activity为DC coupled 或AC coupled模式。
28	THRESH_FF	读/写	0x00	Free-Fall 的门限	0x0C	设置Free-Fall的门限为0.75 g
29	TIME_FF	读/写	0x00	Free-Fall 的时间	0x06	设置Free-Fall的时间为30 ms
2A	TAP_AXES	读/写	0x00	Tap/Double Tap使能控制	不使用	
2B	ACT_TAP_STATUS	只读	0x00	Activity/Tap中断轴指示	只读	
2C	BW_RATE	读/写	0x0A	采样率和功耗模式控制	0x0A	设置采样率为 100 Hz
2D	POWER_CTL	读/写	0x00	工作模式控制	0x00	设置为正常工作模式
2E	INT_ENABLE	读/写	0x00	中断使能控制	0x1C	使能Activity, Inactivity, Free-Fall 中断。
2F	INT_MAP	读/写	0x00	中断影射控制	0x00	M所有中断影射到Int1 管脚
30	INT_SOURCE	只读	0x00	中断源指示	只读	
31	DATA_FORMAT	读/写	0x00	数据格式控制	0x0B	设置为 ± 16 g 测量范围, 13-bit右对齐模式, 中断为高电平触发, 使用I2C 数据接口。
32	DATAx0	只读	0x00	X轴数据	只读	
33	DATAx1	只读	0x00	只读		
34	DATAY0	只读	0x00	Y轴数据	只读	
35	DATAY1	只读	0x00	只读		
36	DATAZ0	只读	0x00	Z轴数据	只读	
37	DATAZ1	只读	0x00	只读		

寄存器0x32至0x37—DATAx0、DATAx1、DATAY0、DATAY1、DATAZ0和DATAZ1(只读)

这6个字节(寄存器0x32至寄存器0x37)都为8位字节, 保存各轴的输出数据。寄存器0x32和0x33保存x轴输出数据, 寄存器0x34和0x35保存y轴输出数据, 寄存器0x36和0x37保存z轴输出数据。输出数据为二进制补码, DATAx0为最低有效字节, DATAx1为最高有效字节, 其中x代表X、Y或Z。DATA_FORMAT寄存器(地址0x31)控制数据格式。建议所有寄存器执行多字节读取, 以防止相继寄存器读取之间的数据变化。

运行结果

```
pi@raspberrypi:~ $ python adxl345_g.py  
ADXL345 on address 0x53: x = -0.672G, y = 0.260G, z = 0.604G  
ADXL345 on address 0x53: x = -0.668G, y = 0.268G, z = 0.612G  
ADXL345 on address 0x53: x = -0.672G, y = 0.272G, z = 0.612G
```


作业2

基于加速度计的跌倒检测系统：

- 要求：当人或物体跌倒时，**LED**闪烁报警
- 上述内容书写报告，将报告、演示视频、实现代码打包上传至Int课程平台，截止时间**2024年3月28日**

- 背景：跌倒本身可能对人体产生伤害，不仅是对老人，在很多其他情况下，跌倒的报警也是非常有帮助的，尤其是从比较高的地方跌倒下来的时候。比如人们在登山，建筑，擦窗户，刷油漆和修理屋顶的时候
- 原理：设计基于三轴加速度传感器的跌倒检测器成为可能。这种跌倒检测器的基本原理是通过测量佩戴该仪器的个体在运动过程中的三个正交方向的加速度变化来感知其身体姿态的变化，并通过算法分析判断该个体是否发生跌倒情况
- 对跌倒检测原理的研究主要是找到人体在跌倒过程中的加速度变化特征

- 失重
- 撞击
- 静止
- 与初始状态比较
- 参考文献:

- <http://www.elecfans.com/d/897069.html>

- 基于 ADXL345 防跌倒检测装置的研制 微型机与应用 2017年第36卷

- <https://max.book118.com/html/2017/0920/134485071.shtm>

