

智能系统与控制

树莓派：时钟模块-ds1302



于泓
鲁东大学
信息与电气工程学院
2021.11.4

树莓派的时间同步

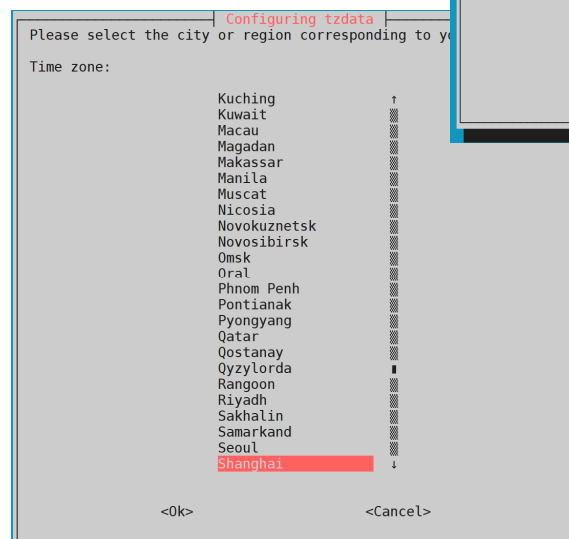
- 树莓派内部，并没有内置时钟模块，在联网状态下，可以依靠网络利用**NTP网络时间协议**来同步时间。

`sudo timedatectl set-ntp true`

```
pi@raspberrypi:~ $ sudo timedatectl set-ntp true
pi@raspberrypi:~ $ date
Thu  4 Nov 08:49:42 CST 2021
```

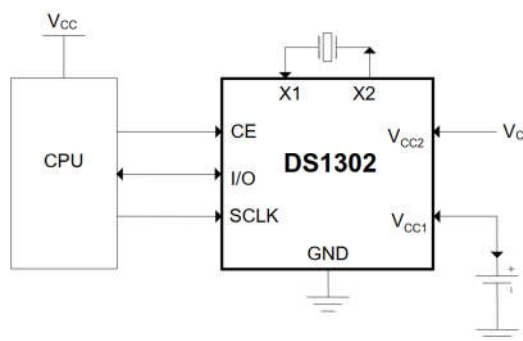
有时会出现时区的错误

`sudo dpkg-reconfigure tzdata` → 修改时区



DS1302 时间模块

- 在没有联网的情况下，需要利用时间模块来获取时间



DS1302是DALLAS(达拉斯)公司出的一款涓流充电时钟芯片，广泛应用于电话、传真、便携式仪器等产品领域，

- 1、可以提供秒、分、小时、日期、月、年等信息，
- 2、拥有31字节数据存储RAM。
- 3、串行I/O通信方式，相对并行来说比较节省IO口的使用。
- 4、DS1302的工作电压比较宽，大概是2.0V~5.5V都可以正常工作。
- 5、DS1302这种时钟芯片功耗一般都很低，它在工作电压2.0V的时候，工作电流小于300nA。

Ds1302 工作时序

设置时间： 写**写地址** 写 数据

读取时间： 写**读地址** 读 数据

地址定义方式

7	6	5	4	3	2	1	0
1	RAM CK	A4	A3	A2	A1	A0	RD WR

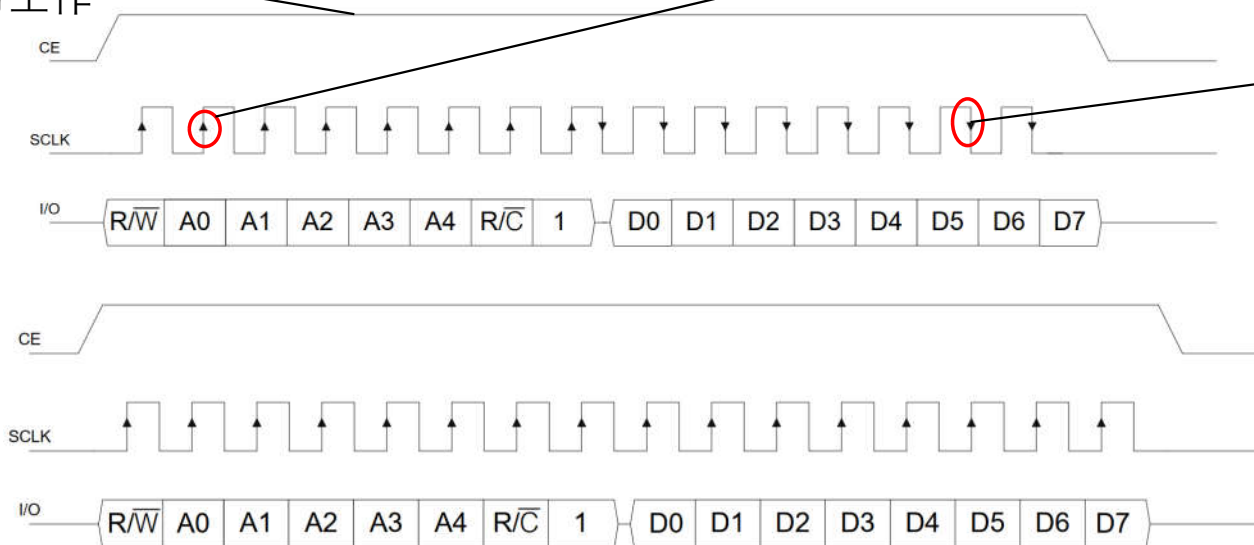
CE 高电平
正常工作

上升沿，写入一个bit

下降沿读取一个bit

读一个字节数据数据

写一个字节数据数据



READ	WRITE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	RANGE
81h	80h	CH		10 Seconds			Seconds			00-59
83h	82h			10 Minutes			Minutes			00-59
85h	84h	12/24	0	10 AM/PM	Hour		Hour			1-12/0-23
87h	86h	0	0	10 Date			Date			1-31
89h	88h	0	0	0	10 Month		Month			1-12
8Bh	8Ah	0	0	0	0	0	Day			1-7
8Dh	8Ch			10 Year			Year			00-99
8Fh	8Eh	WP	0	0	0	0	0	0	0	—
91h	90h	TCS	TCS	TCS	TCS	DS	DS	RS	RS	—

时间寄存器

直接使用24小时制
(12小时可以通过Python程序调整)

星期几

写保护

涓流模式

CLOCK BURST

BFh	BEh
-----	-----

数据连续读写

RAM

C1h	C0h		00-FFh
C3h	C2h		00-FFh
C5h	C4h		00-FFh
.	.		.
.	.		.
.	.		.
FDh	FCh		00-FFh

31字节的RAM

RAM BURST

FFh	FEh
-----	-----

Ds1302 一次数据读写的一般过程

(1) 读写准备

clk=0 dat=0 **rst=0**

延时

clk=0 dat=0 **rst=1**

(2) 写 读/写 地址

(3) 读/写数据

(4) 读写结束

clk=0 dat=0 **rst=0**

代码实现

```
import RPi.GPIO as GPIO
from pin_dic import pin_dic
from datetime import datetime
import operator
import time
```

`operator.mod(Second, 10)` 取余数

`operator.floordiv(Second, 10)` 除法取整数商

`operator.mod(Second, 10) | operator.floordiv(Second, 10) * 16` 将整数变BCD码

`operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10` BCD 变整数

```
>>> from datetime import datetime
>>> dt = datetime(2021,11,4,9,25,23)
>>> print(dt)
2021-11-04 09:25:23
```

```
>>> dt.year
2021
>>> dt.day
4
>>> dt.month
11
```

```
>>> str_dt = dt.strftime("%a %Y-%m-%d %H:%M:%S")
>>> print(str_dt)
Thu 2021-11-04 09:25:23
```

指令	描述	实例	TRY
%a	Weekday, 短版本	Wed	试一试
%A	Weekday, 完整版本	Wednesday	试一试
%w	Weekday, 数字 0-6, 0 为周日	3	试一试
%d	日, 数字 01-31	31	试一试
%b	月名称, 短版本	Dec	试一试
%B	月名称, 完整版本	December	试一试
%m	月, 数字01-12	12	试一试
%y	年, 短版本, 无世纪	18	试一试
%Y	年, 完整版本	2018	试一试
%H	小时, 00-23	17	试一试
%I	小时, 00-12	05	试一试
%p	AM/PM	PM	试一试
%M	分, 00-59	41	试一试
%S	秒, 00-59	08	试一试

https://www.w3school.com.cn/python/python_datetime.asp

2021/11/14

```
class DS1302(object):  
  
    def __init__(self, pin_clk, pin_dat, pin_rst):  
        self.pin_clk = pin_clk  
        self.pin_dat = pin_dat  
        self.pin_rst = pin_rst  
  
        self.period = 0.00001  
        self.init_ds1302()  
  
        # 确定写保护关闭.  
        self.write_byte(int("10001110", 2))  
        self.write_byte(int("00000000", 2))  
  
        # 关闭涓流模式.  
        self.write_byte(int("10010000", 2))  
        self.write_byte(int("00000000", 2))  
        # DS1302 通信终止.  
        self.end_ds1302()
```

```
# DS1302 通信初始化    clk=0 dat=0 rst=1  
def init_ds1302(self):  
    GPIO.setup(self.pin_clk, GPIO.OUT, initial=0)  
    GPIO.setup(self.pin_dat, GPIO.OUT, initial=0)  
    GPIO.setup(self.pin_rst, GPIO.OUT, initial=0)  
  
    GPIO.output(self.pin_clk, 0)  
    GPIO.output(self.pin_dat, 0)  
    time.sleep(self.period)  
    GPIO.output(self.pin_rst, 1)  
  
# DS1302 通信结束    clk=0 dat=0 rst=0  
def end_ds1302(self):  
    GPIO.setup(self.pin_clk, GPIO.OUT, initial=0)  
    GPIO.setup(self.pin_rst, GPIO.OUT, initial=0)  
    GPIO.setup(self.pin_dat, GPIO.OUT, initial=0)  
    GPIO.output(self.pin_clk, 0)  
    GPIO.output(self.pin_dat, 0)  
    time.sleep(self.period)  
    GPIO.output(self.pin_rst, 0)
```



```
# 写入 一个字节
def write_byte(self,Byte):
    # 循环写入每个bit
    for Count in range(8):
        time.sleep(self.period)
        GPIO.output(self.pin_clk, 0)

        # 获取一个bit数据
        Bit = operator.mod(Byte, 2)
        Byte = operator.floordiv(Byte, 2)

        # 将数据送入 io
        time.sleep(self.period)
        GPIO.output(self.pin_dat, Bit)

        # clk 上升沿 将数据写入
        time.sleep(self.period)
        GPIO.output(self.pin_clk, 1)
```

```
# 读取一个字节
```

```
def read_byte(self):
```

```
    # 设置IO为输入
```

```
    GPIO.setup(self.pin_dat, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
    Byte = 0
```

```
    for Count in range(8):
```

```
        # 产生一个上升脉冲
```

```
        time.sleep(self.period)
```

```
        GPIO.output(self.pin_clk, 1)
```

```
        time.sleep(self.period)
```

```
        GPIO.output(self.pin_clk, 0)
```

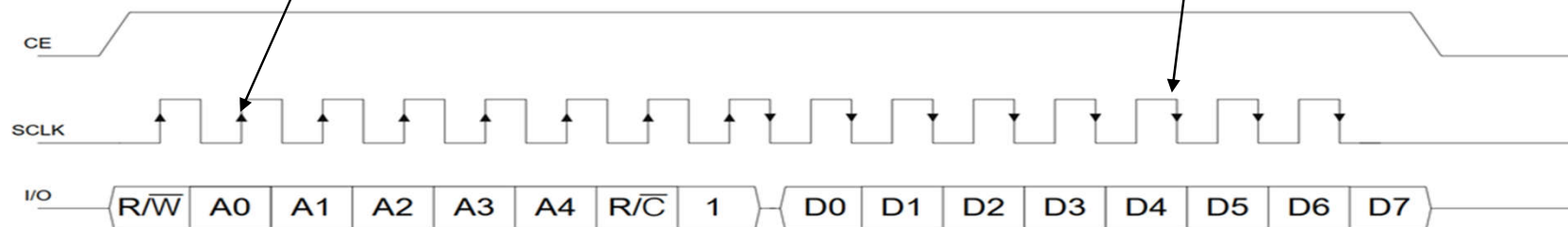
```
        time.sleep(self.period)
```

```
        # 读取1bit数据
```

```
        Bit = GPIO.input(self.pin_dat)
```

```
        Byte |= ((2 ** Count) * Bit)
```

```
    return Byte
```



```
# 向DS1302的RAM里面写数据
def write_RAM(self, Data):
    # DS1302 通信初始化.
    self.init_ds1302()
    # 写地址.
    self.write_byte(int("11111110", 2))
    # 写数据共有31个字节的RAM空间.
    for Count in range(len(Data)):
        self.write_byte(ord(Data[Count:Count + 1]))

    # 剩余部分写空格
    for Count in range(31 - len(Data)):
        self.write_byte(ord(" "))
    # 通信结束.
    self.end_ds1302()
```

RAM BURST

FFh	FEh
-----	-----

连续读写方式

```
# 从 ds1302的RAM 中读取数据
def ReadRAM(self):

    # 初始化 DS1302 通信.
    self.init_ds1302()
    # 写地址.
    self.write_byte(int("11111111", 2))
    # 读数据.
    Data = ""
    for Count in range(31):
        Byte = self.read_byte()
        Data += chr(Byte)
    # 通信结束.
    self.end_ds1302()
    return Data
```

```
# 向DS1302 中写时间
def write_DateTime(self,dt):
    # DS1302 通信初始化.
    self.init_ds1302()

    # 写地址.
    self.write_byte(int("10111110", 2))

    # 写秒.
    Second = dt.second
    self.write_byte(operator.mod(Second, 10) | operator.floordiv(Second, 10) * 16)

    # 写分.
    Minute = dt.minute
    self.write_byte(operator.mod(Minute, 10) | operator.floordiv(Minute, 10) * 16)

    # 写小时.
    Hour = dt.hour
    self.write_byte(operator.mod(Hour, 10) | operator.floordiv(Hour, 10) * 16)

    # 写日期.
    Day = dt.day
    self.write_byte(operator.mod(Day, 10) | operator.floordiv(Day, 10) * 16)

    # 写月.
    Month = dt.month
    self.write_byte(operator.mod(Month, 10) | operator.floordiv(Month, 10) * 16)

    # 写星期.
    DayOfWeek = int(dt.strftime("%w"))
    if DayOfWeek == 0:
        DayOfWeek = 7
    self.write_byte(operator.mod(DayOfWeek, 10) | operator.floordiv(DayOfWeek, 10) * 16)
```

```
# 写年.
Year = dt.year
Year = operator.mod(Year, 100)
self.write_byte(operator.mod(Year, 10) | operator.floordiv(Year, 10) * 16)

# 写保护关闭.
self.write_byte(int("00000000", 2))
# 涓流充电模式关闭.
self.write_byte(int("00000000", 2))
# End DS1302 communication.
self.end_ds1302()
```

保留后2位

datetime 星期 0-6

```
def read_DateTime(self):
```

```
    # DS1302 通信初始化.
```

```
    self.init_ds1302()
```

```
    # 写地址.
```

```
    self.write_byte(int("10111111", 2))
```

```
    # 读秒
```

```
    Byte = self.read_byte()
```

```
    second = operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10
```

```
    # 读分
```

```
    Byte = self.read_byte()
```

```
    minute = operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10
```

```
    # 读小时
```

```
    Byte = self.read_byte()
```

```
    hour = operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10
```

```
    # 读日
```

```
    Byte = self.read_byte()
```

```
    day = operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10
```

```
    # 读月
```

```
    Byte = self.read_byte()
```

```
    month = operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10
```

```
    # 读星期 不用操作
```

```
    Byte = self.read_byte()
```

```
    # 读年
```

```
    Byte = self.read_byte()
```

```
    year = operator.mod(Byte, 16) + operator.floordiv(Byte, 16) * 10
```

```
    year = year+2000
```

```
    self.end_ds1302()
```

数据校验

```
if year == 2000 or month > 12 or month<1 or day < 1 or day > 31:  
    return False
```

```
if second > 59:  
    return False
```

```
return datetime(year,month,day,hour,minute,second)
```

```
if __name__ == "__main__":

    pin_clk = pin_dio['G4']
    pin_dat = pin_dio['G5']
    pin_rst = pin_dio['G6']

    GPIO.setmode(GPIO.BOARD)

    m_ds1302 = DS1302(pin_clk, pin_dat, pin_rst)
    write_dt = datetime(2021, 11, 3, 10, 25, 00)
    m_ds1302.write_DateTime(write_dt)

    try:
        while True:
            dt = m_ds1302.read_DateTime()

            if not dt:
                continue
            else:
                str_time = dt.strftime("%a %Y-%m-%d %H:%M:%S")
                print("\r%s"%(str_time), end="")

                time.sleep(1)

    except KeyboardInterrupt:
        print('\n Ctrl + C QUIT')

    finally:

        GPIO.cleanup()
```