

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



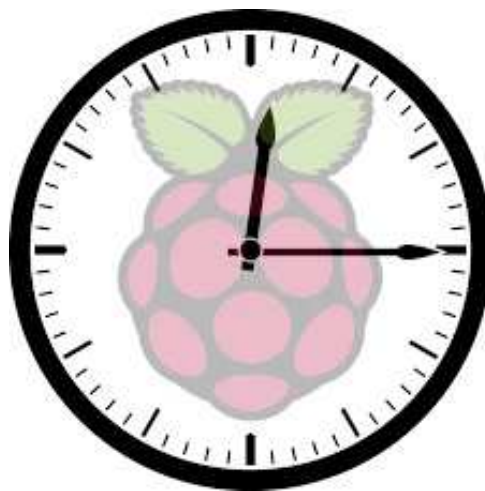
Codul sursa din acest document este licentiat

Public-Domain

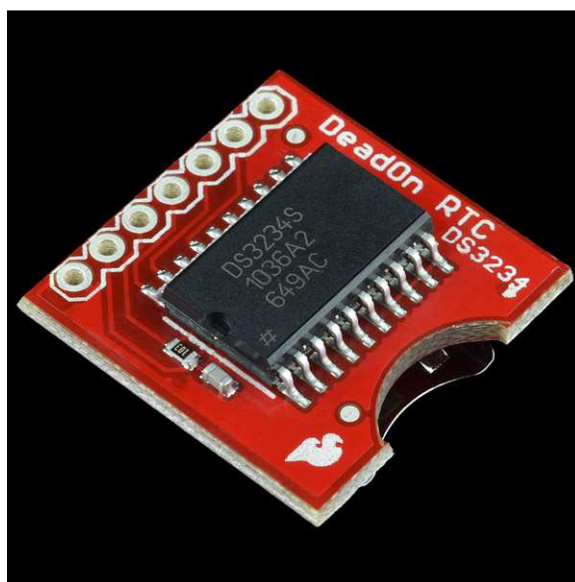
Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

# Real Time Pi 2

În lipsa unei conexiuni Internet un sistem bazat pe Raspberry Pi este incapabil să mențină ora și data exactă dacă se întrerupe alimentarea cu energie electrică. Cea mai simplă modalitate de a rezolva acest lucru este adăugarea unui modul RTC la placa (așa cum am arătat și în lecția precedentă). În cadrul acestei lecții vom prezenta o modalitate de a transforma un sistem Raspberry Pi în sursă de timp într-o rețea de sisteme fără conexiune Internet. Imaginați-vă o rețea de plăci Raspberry Pi, fără conectivitate Internet, care au nevoie de oră și dată exacte și, mai mult decât atât, acestea să fie sincronizate. Instalarea a câte un modul RTC pe fiecare sistem va rezolva problema orei și datei pe fiecare sistem în parte dar nu va asigura sincronizarea exactă între sisteme. Din acest motiv o soluție mult mai elegantă este instalarea unui singur modul RTC și sincronizarea prin rețea a celorlalte sisteme cu sistemul pe care este instalat modulul RTC.



Ca modul RTC vom utiliza un modul Sparkfun DeadOn RTC bazat pe circuitul DS3234 – circuit RTC extrem de precis și care are un sistem de compensare a frecvenței de ceas pentru variații de temperatură.

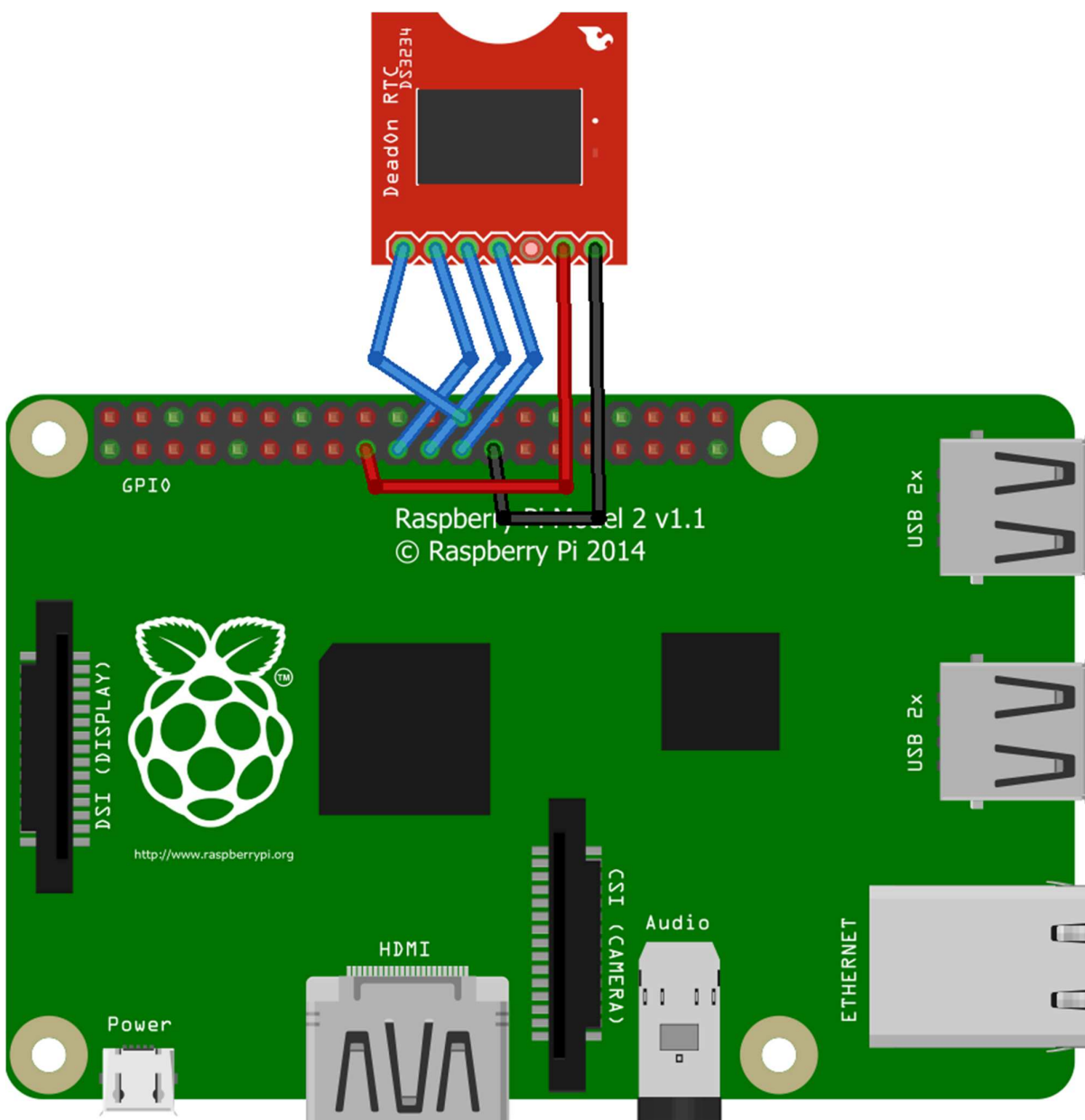


[https://www.robofun.ro/module/module-rtc/real\\_time\\_clock\\_DS3234](https://www.robofun.ro/module/module-rtc/real_time_clock_DS3234)

<https://www.robofun.ro/forum/>

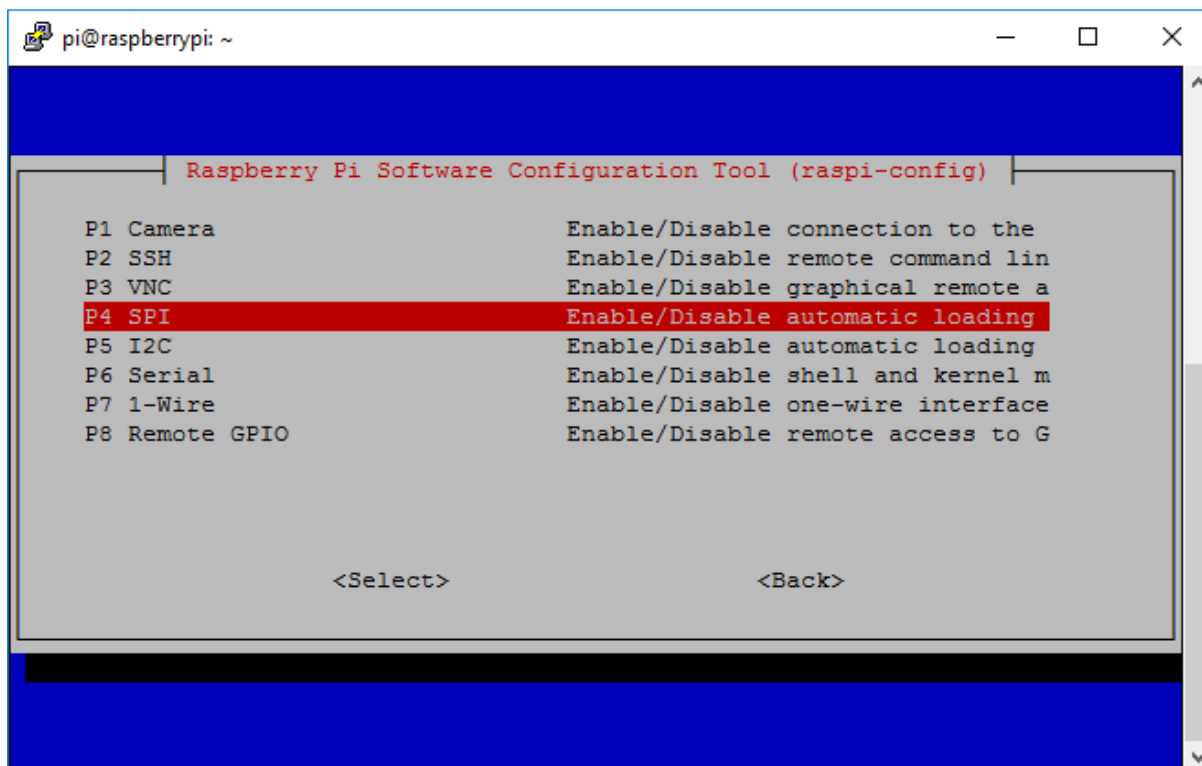
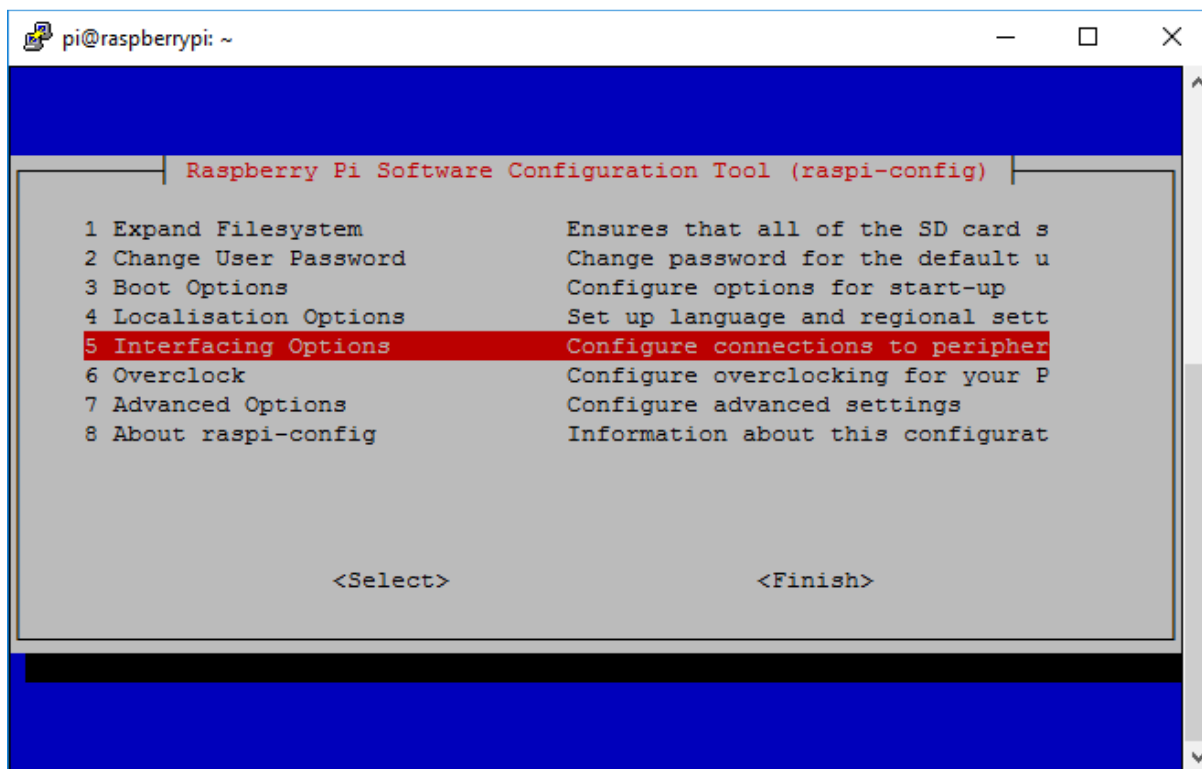
Modulul RTC necesită o baterie de 3V / 12mm (CR1220 de exemplu) pentru a menține ora și data în lipsa alimentării cu energie electrică a sistemului – bateria nu este inclusă la cumpărare modului.

Interconectarea cu placa Raspberry Pi se face prin intermediul magistralei seriale SPI și necesită următoarele conexiuni: pinul SS al modului se va conecta la pinul GPIO8, pinul MOSI la pinul GPIO10, pinul MISO la pinul GPIO9, pinul SCLK la pinul GPIO11, pinul VCC la 3.3V și pinul GND la GND (precum în schema următoare).



Pentru a comunica cu modulul este necesar să activăm comunicația SPI prin intermediul utilitarului *raspi-config*:

```
sudo raspi-config
```





După activarea comunicației SPI vom crea un fișier *ds3234-rpi-overlay.dts* în care vom introduce următoarele:

```
/dts-v1/;
/plugin/;

/ {
    compatible = "brcm,bcm2708";
    fragment@0 {
        target = <&spi0>;
        __overlay__ {
            spidev@0 {
                status = "disabled";
            };
            ds3234@0 {
                compatible = "ds3234";
                reg = <0>;
            };
        };
    };
};
```



După repornire vom verifica încărcarea modulului asociat RTC-ului:

```
pi@raspberrypi ~ $ lsmod | grep ds3234
rtc_ds3234                2088  0
```

și inițializarea corectă a acestuia:

```
pi@raspberrypi ~ $ dmesg | grep ds3234
[4.443480] ds3234 spi0.0: Control Reg: 0x1c
[4.443617] ds3234 spi0.0: Ctrl/Stat Reg: 0x88
[4.456070] ds3234 spi0.0: rtc core: registered ds3234 as rtc0
```

Din acest moment sistemul va utiliza modulul RTC ca sursă de timp. Inițializarea acestuia se va face cu ajutorul comenzii de scriere în memoria internă (sincronizarea de timp inițială):

```
sudo hwclock -w
```

iar verificarea se poate face cu ajutorul comenzii:

```
sudo hwclock -r
```

Pentru ca ora și data sistemului să fie disponibile în rețea este necesară configurarea unui server de NTP (Network Time Protocol). În general pachetul *ntp* este deja instalat în distribuția Linux a plăcii Raspberry Pi dar funcționează ca și client (sincronizează ceasul local prin intermediul rețelei Internet utilizând servere NTP publice). Pentru a activa caracteristica de server este necesară editarea fișierului */etc/ntp.conf*, comentarea tuturor liniilor și înscrierea următoarei configurații (sau se poate înlocui fișierul cu totul):

```
driftfile /var/lib/ntp/ntp.drift
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
```

```
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
```

Pentru activarea propriu-zisă a componentei de server sunt necesare următoarele linii (15 este rangul Stratum al serverului NTP, în cazul nostru este extrem de scăzut fiindcă sincronizarea nu se bazează pe un server extern):

```
server 127.127.1.0
fudge 127.127.1.0 stratum 15
```

Următoarele două linii sunt necesare pentru ca celelalte sisteme să poată interoga serverul (IPv4 și IPv6):

```
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
```

Vom permite interogările de pe sistemul local:

```
restrict 127.0.0.1
restrict ::1
```

și din rețeaua locală din care face parte sistemul (se presupune că este o clasă C de tipul *192.168.xxx.0/24*):

```
restrict 192.168.xxx.0 mask 255.255.255.0 notrap
broadcast 192.168.xxx.255
```

După salvarea fișierului se va reporni serviciul ntp:

```
sudo /etc/init.d/ntp restart
```

și vom verifica starea serverului NTP:



```
pi@raspberrypi /etc $ ntpq -c rv 127.0.0.1
associd=0 status=0515 leap_none, sync_local, 1 event, clock_sync,
version="ntpd 4.2.6p5@1.2349-o Mon Jul 25 22:35:28 UTC 2016 (1)",
processor="armv7l", system="Linux/4.4.38-v7+", leap=00, stratum=16,
precision=-20, rootdelay=0.000, rootdisp=10.927, refid=...,
reftime=dc06888a.0d4b5241 Thu, Dec 22 2016 19:14:50.051,
clock=dc06888a.405fc437 Thu, Dec 22 2016 19:14:50.251, peer=55790,
tc=6,
mintc=3, offset=0.000, frequency=-5.806, sys_jitter=0.000,
clk_jitter=0.001, clk_wander=0.000
```

Pe celelalte sisteme din rețea (sistemele client) se va edita același fișier */etc/ntp.conf*, se va configura o directivă server cu adresa sistemului configurat ca server NTP (celelalte servere se pot comenta) și se va reporni serviciul *ntp*.

Configurația prezentată a fost testată pe sisteme rulând Raspbian GNU/Linux 8 (jessie), kernel 4.4.38-v7+ și ntp 4.2.6p5.

Materialul de față a fost inspirat de următoarele proiecte:

Raspberry PI and your SPI Real Time Clock (RTC)

<http://www.sciencegizmo.com.au/?p=137>

The Raspberry Pi as a Stratum-1 NTP Server

<http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html>