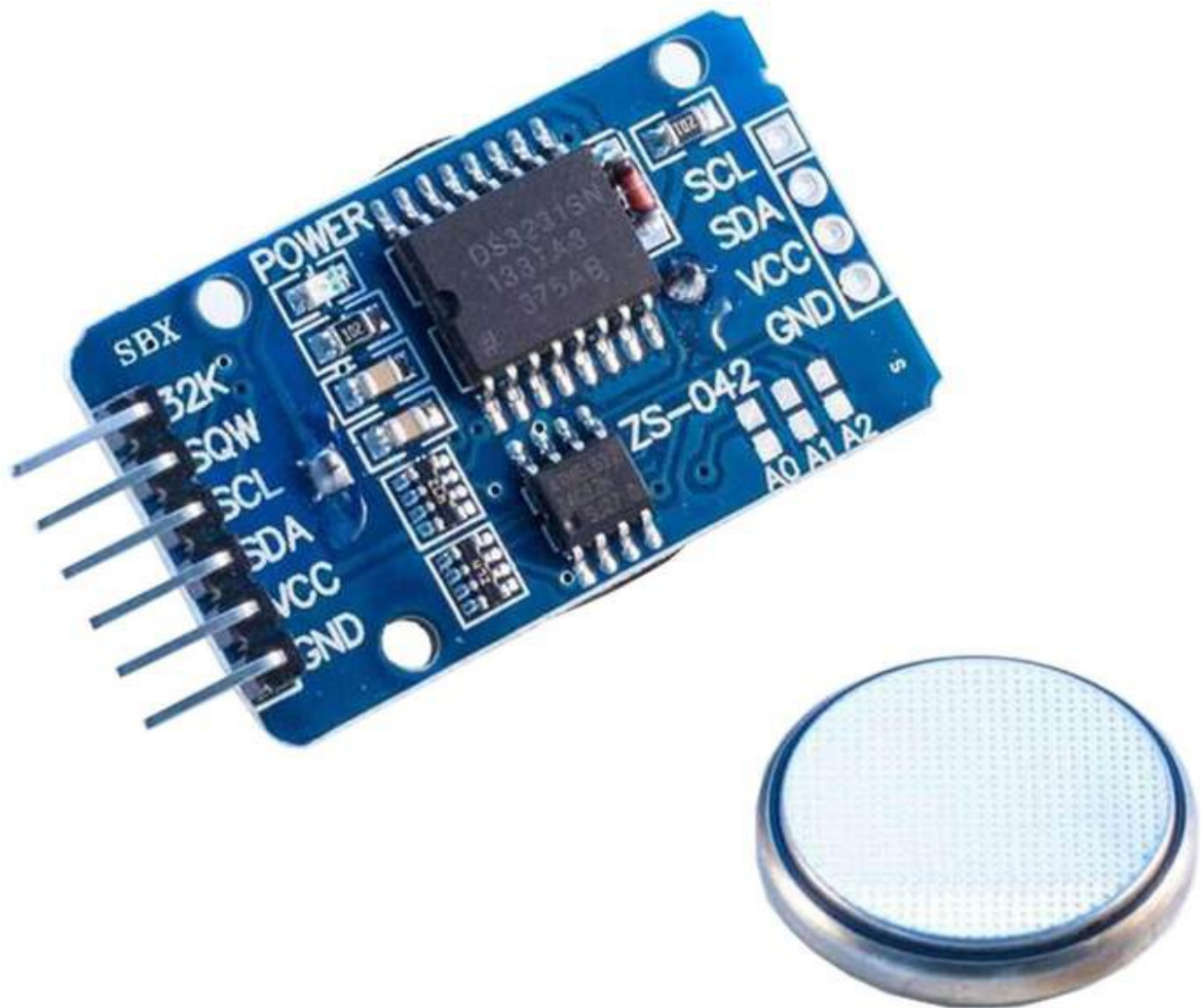


Az-Delivery

Benvenuto!

Grazie per aver acquistato il nostro *Modulo DS3231 Orologio Real-Time AZ-Delivery*. Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!





Indice dei Contenuti

Introduzione.....	3
Specifiche.....	5
La piedinatura.....	6
Come configurare l'Arduino IDE.....	7
Come configurare il Raspberry Pi e il Python.....	11
Collegamento del modulo con Uno.....	12
Esempio di sketch.....	14
Collegamento del modulo con Raspberry Pi.....	20
Abilitazione dell'interfaccia I2C.....	21
Librerie e strumenti per Python.....	23
Script Python.....	24



Introduzione

Il modulo Orologio in Tempo Reale DS3231 viene utilizzato come dispositivo di sincronizzazione temporale in applicazioni in cui sono essenziali tempi precisi. Il modulo è usato in orologi digitali, schede madri di computer, macchine fotografiche digitali, sistemi incorporati ecc..

È un orologio in tempo reale con un oscillatore a cristalli a temperatura compensata integrato. C'è un supporto per la batteria a bordo, in modo che possa mantenere l'ora quando il dispositivo non è alimentato da una fonte esterna.

Una delle caratteristiche del modulo è che può funzionare in formato 12 ore o 24 ore e ha una capacità di indicazione AM/PM.

Il modulo è programmabile con due allarmi diurni. Gli allarmi possono essere programmati tramite un chip EEPROM integrato che può memorizzare i dati degli allarmi nella memoria interna. C'è anche un pin di uscita dell'oscillatore a onda quadra 32KHz, che può essere usato per sincronizzare l'ora con altri dispositivi simili.

Az-Delivery

L'orologio interno può fornire informazioni su secondi, minuti, ore, giorno, data, mese e anno. La data alla fine del mese è automaticamente regolata per i mesi che hanno meno di 31 giorni. Include anche le correzioni per gli anni bisestili.

Il modulo ha un'interfaccia I2C con indirizzo seriale I2C, e può essere collegato insieme ad altri dispositivi sulle stesse linee I2C.

Specifiche

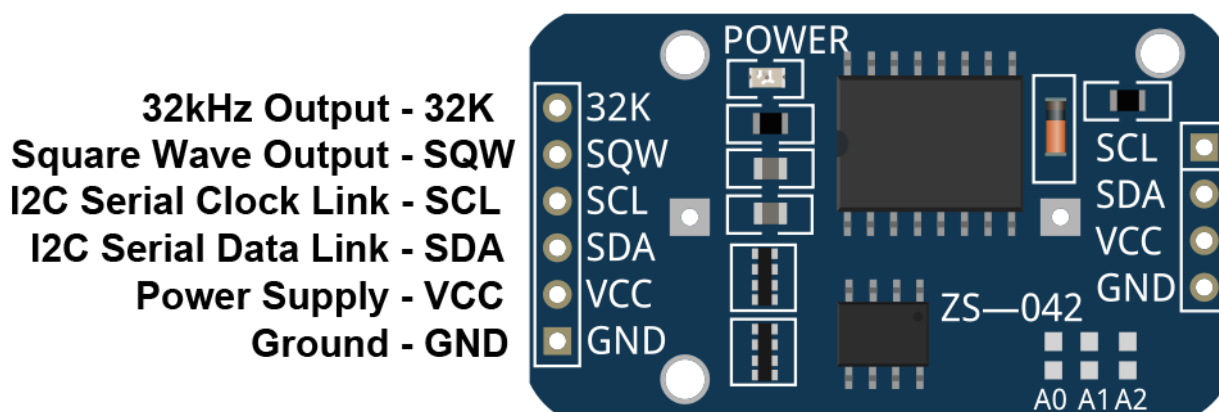
Tensione di alimentazione	da 3.3V a 5V
Temperatura di esercizio	da 0°C a +70°C
Interfaccia di comunicazione	I2C
Batteria di backup	Un supporto per batteria a bottone da 3V
Sensore di temperatura digitale	Precisione $\pm 3^{\circ}\text{C}$
Onda quadra programmabile	32kHz [Output]
Orario degli allarmi giornalieri	2
Basso Consumo Energetico	meno di 1mA
Dimensioni	34 x 23 x 18mm [1.3 x 09 x 07in]

Il modulo consiste in un chip RTC Clock DS3231 e un chip EEPROM Atmel AT24C32. L'AT24C32 ha una capacità di memoria di 32kB e utilizza l'interfaccia bus I2C con indirizzo 0x57 che può essere modificato. Ha la capacità di impostare l'ora e la data, controllare e cancellare gli allarmi e registrare i dati con un timestamp.

Il modulo ha un supporto per una batteria a bottone da 3V e la batteria è inclusa nel modulo. La batteria serve come alimentazione di riserva per il modulo. Quando l'alimentazione esterna è spenta, il chip di rilevamento automatico a bordo passa alla batteria di riserva.

La piedinatura

Il modulo RTC DS3231 ha sei pin su un lato e altri quattro, per l'alimentazione e le linee di interfaccia I2C sull'altro lato. La piedinatura è mostrata nell'immagine seguente:



Il modulo RTC DS3231 funziona in modo sicuro nella gamma da 3,3V a 5V.

Il pin di uscita 32K è un pin di uscita dell'oscillatore controllato da un cristallo. Fornisce un segnale a onda quadra di 32kHz e può essere utilizzato per alimentare il segnale di riferimento per altri dispositivi. Può essere lasciato galleggiare se non utilizzato.

Il pin SQW può fornire sia un segnale di interrupt dovuto a condizioni di allarme che un segnale di uscita a onda quadra.

Come configurare l'Arduino IDE

Se l'Arduino IDE non è installato, seguire il [link](#) e scaricare il file di installazione del sistema operativo scelto.

Download the Arduino IDE



ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#) 

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Per gli utenti *Windows*, fare doppio clic sul file `.exe` scaricato e seguire le istruzioni nella finestra di installazione.

Az-Delivery

Per gli utenti Linux, scaricare un file con estensione *.tar.xz*, che è necessario estrarre. Quando lo si estrae, andare nella directory estratta, e aprire il terminale in quella directory. È necessario eseguire due *script .sh*, il primo chiamato *arduino-linux-setup.sh* e il secondo chiamato *install.sh*.

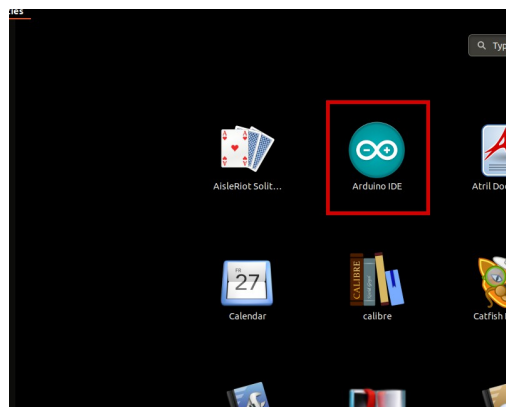
Per eseguire lo script, aprire il terminale nella directory in cui è stato salvato lo script ed eseguire il seguente comando:

sh arduino-linux-setup.sh user_name

user_name - è il nome di un superutente nel sistema operativo Linux.. All'avvio del comando deve essere inserita una password per il superutente. Aspettate qualche minuto che lo script completi tutto.

Il secondo script, chiamato *install.sh*, deve essere usato dopo l'installazione del primo script. Eseguire il seguente comando nel terminale (directory estratta): **sh install.sh**

Dopo l'installazione di questi script, andare su *Tutte le App*, dove troverai l'*Arduino IDE* installato.



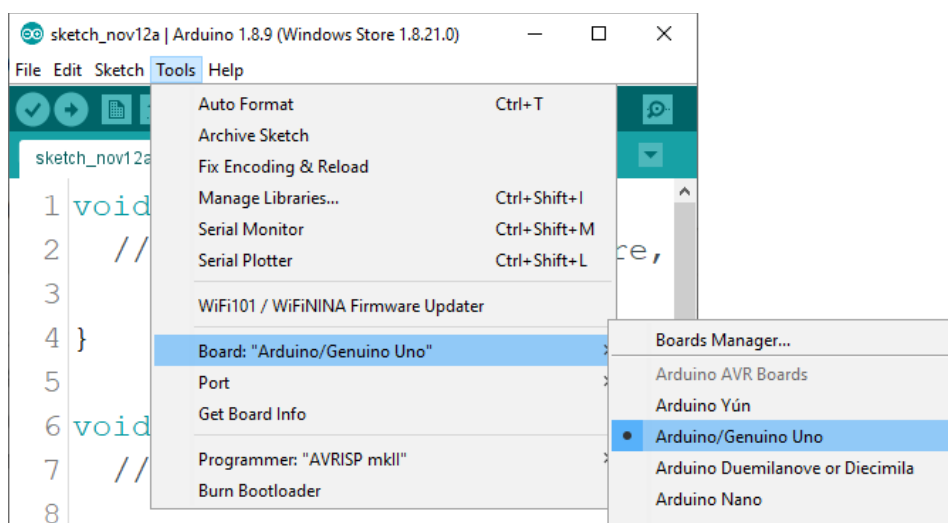
Az-Delivery

Quasi tutti i sistemi operativi sono dotati di un editor di testo preinstallato (ad esempio *Windows* viene fornito con *Notepad*, *Linux Ubuntu* viene fornito con *Gedit*, *Linux Raspbian* viene fornito con *Leafpad*, ecc.). Tutti questi editor di testo sono perfettamente adatti allo scopo dell'eBook.

La prossima cosa da fare è controllare se il PC è in grado di rilevare la scheda Arduino. Aprite l'Arduino IDE appena installato e andate su:

Strumenti > Scheda > {your board name here}

{your board name here} dovrebbe essere l'*Arduino/Genuino Uno*, come si può vedere nella seguente immagine:

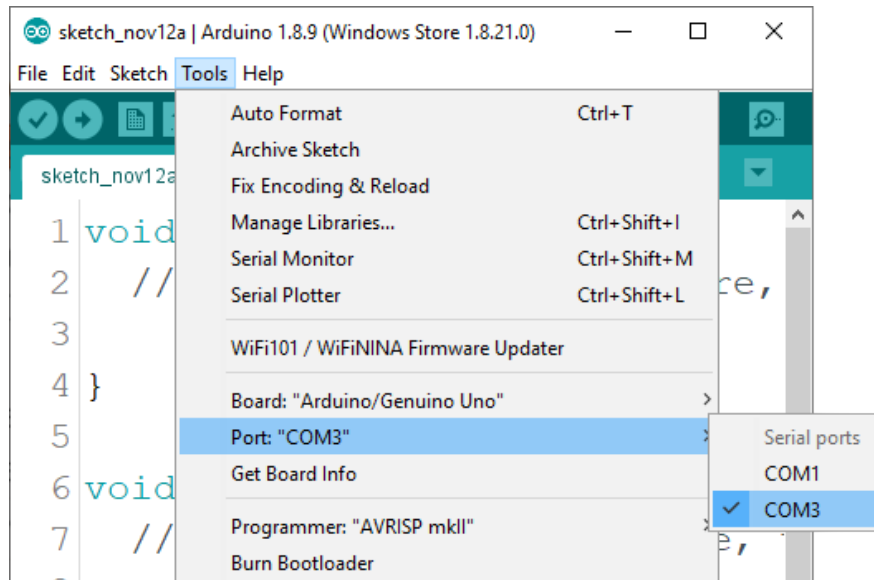


Deve essere selezionata la porta alla quale è collegata la scheda Arduino.

Vai su: *Strumenti > Porta > {port name goes here}*

e quando la scheda Arduino è collegata alla porta USB, il nome della porta è visibile nel menu a tendina dell'immagine precedente.

Se si utilizza l'Arduino IDE su Windows, i nomi delle porte sono i seguenti:



Per gli utenti Linux, il nome della porta è `/dev/ttyUSBx` per esempio, dove x rappresenta un numero intero compreso tra 0 e 9.



Come configurare il Raspberry Pi e il Python

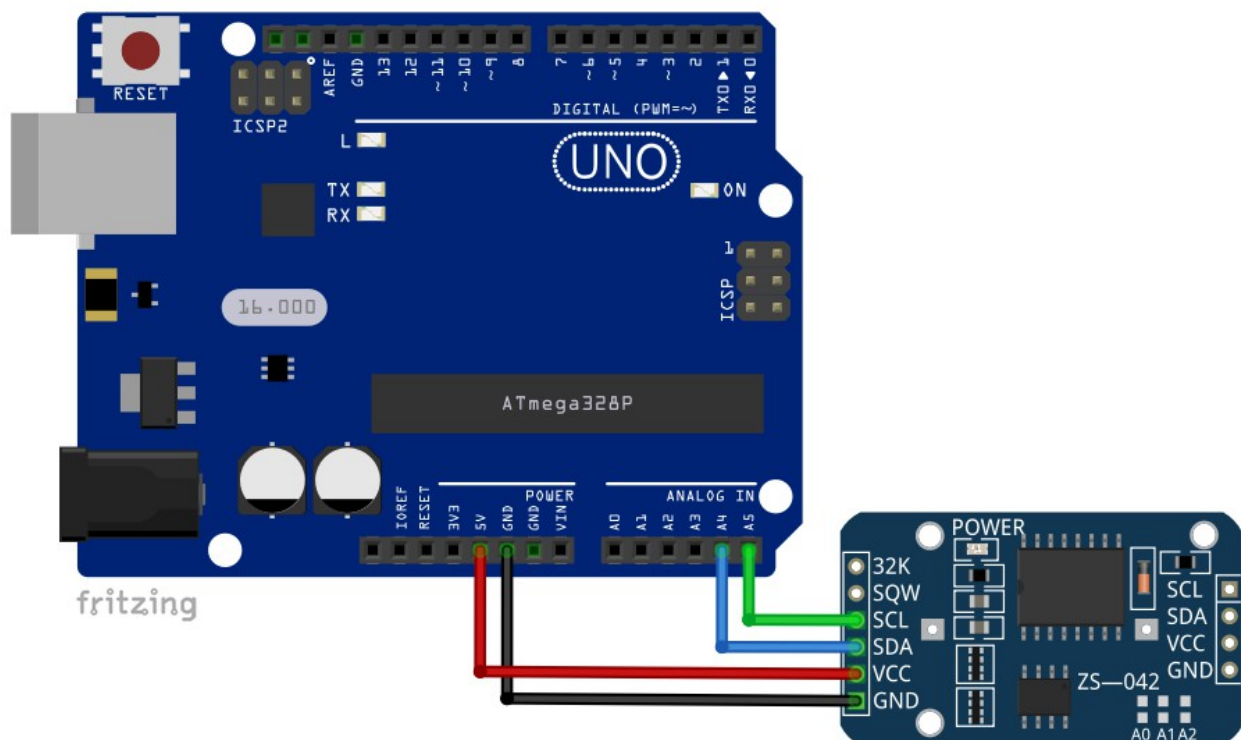
Per il Raspberry Pi, prima deve essere installato il sistema operativo, tutto deve essere impostato in modo da poter essere utilizzato in modalità Headless. La modalità Headless consente la connessione remota al Raspberry Pi, senza la necessità di uno schermo di un PC, mouse o tastiera. Le uniche cose di cui avete bisogno per questa modalità sono il Raspberry Pi, l'alimentazione e la connessione internet. Tutto questo è spiegato in dettaglio nell'eBook gratuito:

[Raspberry Pi Quick Startup Guide](#)

Il sistema operativo Raspbian viene fornito con il Python preinstallato.

Collegamento del modulo con Uno

Collegare il modulo RTC DS3231 con Uno come mostrato nel seguente schema di collegamento:



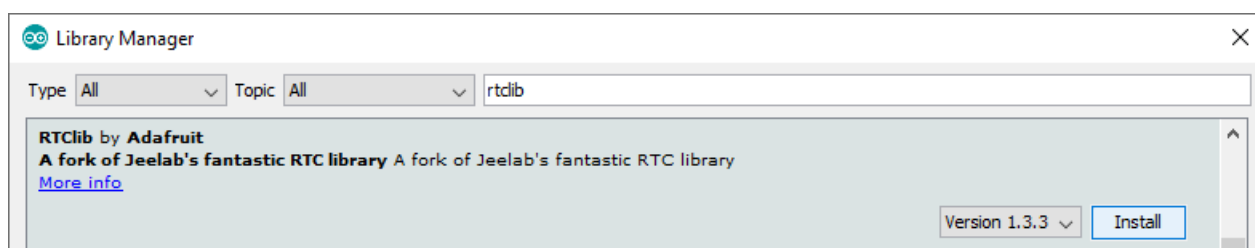
Pin modulo RTC	Pin Uno	Colore filo
SCL	5V	Filo verde
SDA	A5	Filo blu
VCC	A4	Filo rosso
GND	GND	Filo nero

Libreria per Arduino IDE

Per utilizzare il modulo con Uno, si raccomanda di scaricare una libreria esterna. La libreria che useremo si chiama *RTClib*. La versione della libreria utilizzata è la 1.3.3. Per scaricarla e installarla, aprire l'Arduino IDE e andare su:

Strumenti > Gestione Librerie.

Quando si apre una nuova finestra, digitare *RTClib* nella casella di ricerca e installare la libreria *RTClib* fatta da *Adafruit*, come mostrato nella seguente immagine:



Con la libreria vengono forniti diversi esempi di sketch, per aprirne uno, andate su:

File > Esempi > RTClib > ds3231

Grazie a questo esempio di sketch è possibile testare il modulo. Lo sketch in questo eBook è una versione modificata di questo sketch, per ottenere un output più facile da usare.

Az-Delivery

Esempio di sketch

```
#include <Wire.h>
#include "RTClib.h"
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {
  "Sunday",
  "Monday",
  "Tuesday",
  "Wednesday",
  "Thursday",
  "Friday",
  "Saturday"
};
void setup () {
  Serial.begin(9600);
  delay(2000);
  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  /* To manually set date and time,
   remove the coment // signs
   and enter new values in the followingline
   in this sequence: year, day, month, hour, minute and second.*/
  //rtc.adjust(DateTime(2020, 2, 24, 10, 00, 0));
}
void loop () {
  DateTime now = rtc.now();
  //Day of the week
  Serial.print("Day of the week: ");
  Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
  Serial.println();
}
```

Az-Delivery

```
//one tab
//Current time:
Serial.print("Current Time: ");
if (now.hour() < 10) {
    Serial.print("0");
    Serial.print(now.hour());
}
else {
    Serial.print(now.hour(), DEC);
}
Serial.print(':');

if (now.minute() < 10) {
    Serial.print("0");
    Serial.print(now.minute());
}
else {
    Serial.print(now.minute(), DEC);
}
Serial.print(':');
if (now.second() < 10) {
    Serial.print("0");
    Serial.print(now.second());
}
else {
    Serial.print(now.second(), DEC);
}
Serial.println();
```

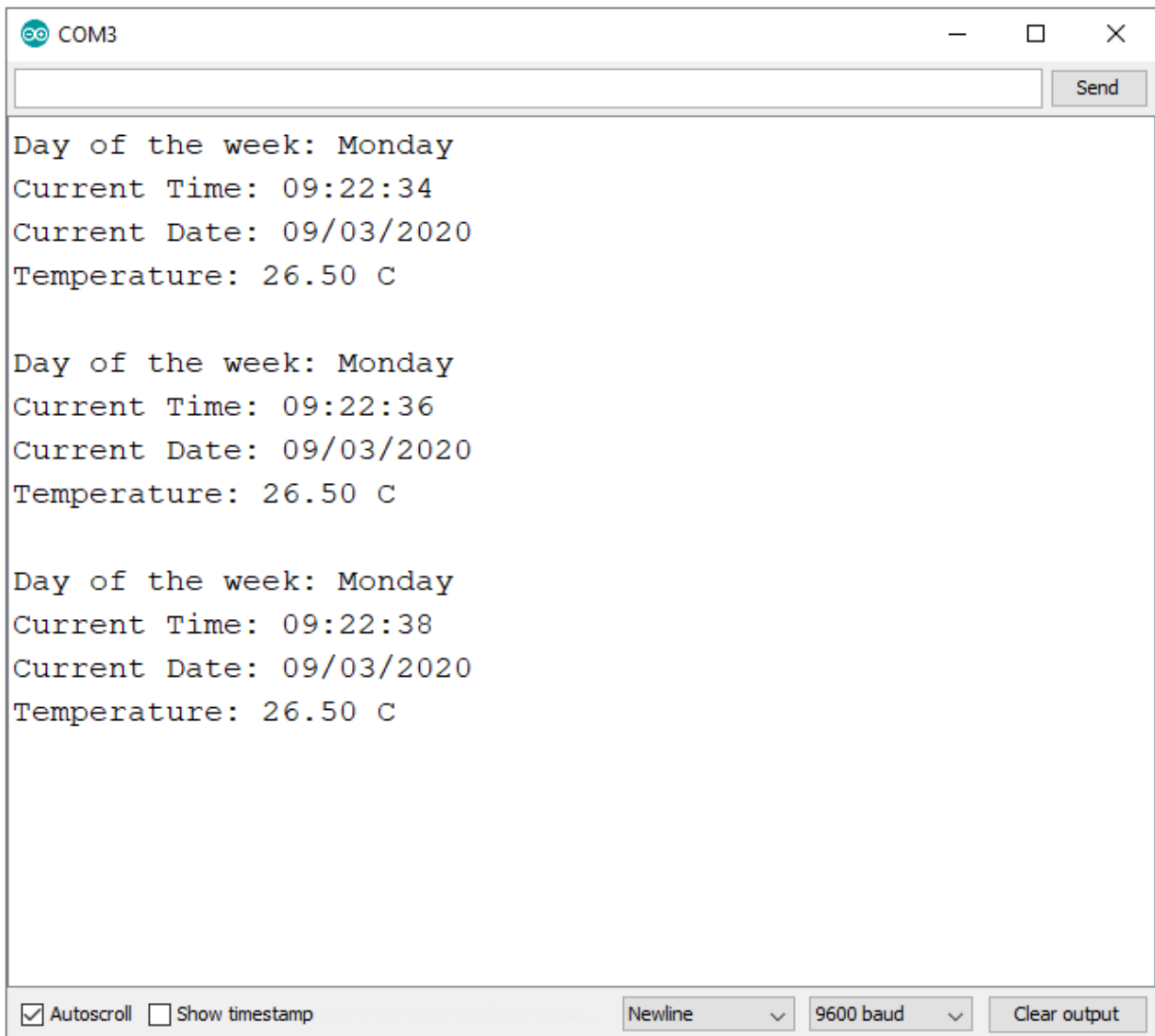
Az-Delivery

```
//one tab
//Current date:
Serial.print("Current Date: ");
if (now.day() < 10) {
    Serial.print("0");
    Serial.print(now.day());
}
else {
    Serial.print(now.day(), DEC);
}
Serial.print('/');
if (now.month() < 10) {
    Serial.print("0");
    Serial.print(now.month());
}
else {
    Serial.print(now.month(), DEC);
}
Serial.print('/');
Serial.print(now.year(), DEC);
Serial.print("");
Serial.println();
//Temperature:
Serial.print("Temperature: ");
Serial.print(rtc.getTemperature());
Serial.println(" C");
Serial.println();
delay(2000);
}
```


Az-Delivery

Caricare lo sketch su Uno e aprire il Monitor Seriale (*Strumenti > Monitor Seriale*).

Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:



Az-Delivery

All'inizio dello sketch vengono importate due librerie chiamate *Wire* e *RTClib*. Queste librerie sono utilizzate per importare funzioni che possono essere utilizzate per la comunicazione tra il modulo e Uno.

Successivamente, viene creato un oggetto chiamato *RTC_DS3231* con la seguente linea di codice: *RTC_DS3231 rtc;*

Dove l'oggetto *rtc* rappresenta il modulo RTC.

Poi viene creato un array di caratteri chiamato *daysOfTheWeek* che rappresenta i nomi dei giorni della settimana, e questi nomi predefiniti sono utilizzati per visualizzare lo stato del giorno corrente nel Monitor Seriale.

All'inizio della funzione *setup()*, si avvia la comunicazione tra Uno e RTC con la seguente linea di codice: *Rtc.Begin();*

Successivamente, viene utilizzata la funzione chiamata *rtc.adjust()*. Questa funzione imposta la data e l'ora nel modulo. Ha un argomento, un oggetto *DateTime*. Per impostare la data e l'ora attuali del PC durante il caricamento, si usa la seguente linea di codice:
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

Az-Delivery

Quando è necessario impostare un'ora e una data specifiche, lo si può fare con le funzioni `adjust()` e `DateTime()`. Per esempio, se si vuole impostare la data su: 09 marzo 2020. E l'ora su 10:00:00, si può usare la seguente riga di codice: `rtc.adjust(DateTime(2020, 3, 09, 10, 00, 0));`

La funzione `DateTime()` ha sei argomenti e restituisce un oggetto `DateTime`. Gli argomenti sono numeri interi, che rappresentano data e ora, nella seguente sequenza: anno, giorno, mese, ora, minuto e secondo. Il valore restituito è un oggetto `DateTime` necessario per impostare la data e l'ora nel modulo.

All'inizio della funzione `loop()`, vengono letti i dati dal modulo e vengono memorizzate le informazioni nell'oggetto chiamato ora con la seguente linea di codice: `DateTime now = rtc.now();`

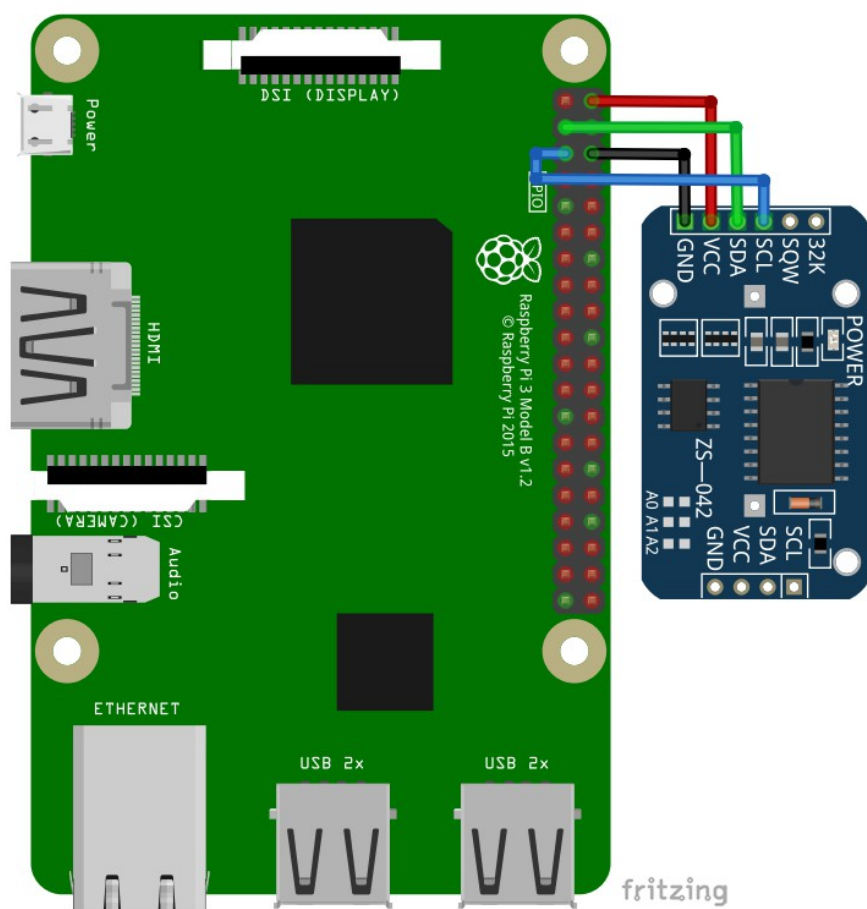
L'oggetto ora è di tipo `DateTime`. Dopo aver letto e memorizzato i dati in questo oggetto, le proprietà di questo oggetto vengono richiamate per visualizzare i dati.

Il seguente è l'algoritmo per la visualizzazione dei dati.

In primo luogo, viene stampato il giorno della settimana. Poi, l'ora corrente. L'output dello sketch originale mostrava valori senza uno zero iniziale, quindi sono state aggiunte semplici istruzioni `if` e `else` per un output migliore. Ogni volta che i valori numerici sono inferiori a 10, viene aggiunto uno zero iniziale all'output.

Collegamento del modulo con Raspberry Pi

Collegare il modulo RTC DS3231 con il Raspberry Pi come mostrato nel seguente sketch di collegamento:

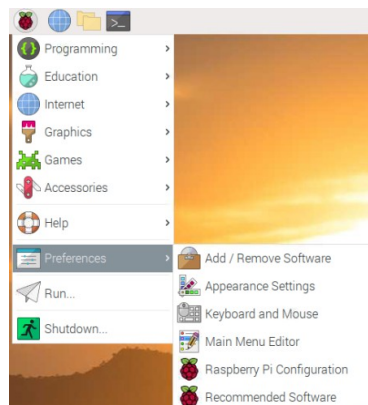


Pin RTC	Pin Raspberry Pi	Pin fisico	Colore filo
VCC	3V3	1	Filo rosso
SDA	GPIO2	3	Filo verde
GND	GND	6	Filo nero
SCL	GPIO3	5	Filo blu

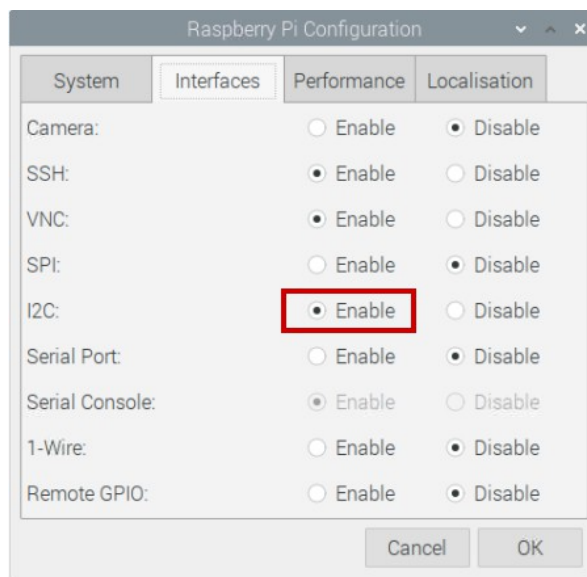
Abilitazione dell'interfaccia I2C

Per utilizzare il modulo con Raspberry Pi, bisogna abilitare l'interfaccia I2C del Raspberry Pi. Aprite il seguente menu:

Menu Applicazione > Preferiti > Configurazione Raspberry Pi



Nella nuova finestra, sotto la scheda *Interfacce*, abilitare il radio button I2C, come nell'immagine seguente:



Az-Delivery

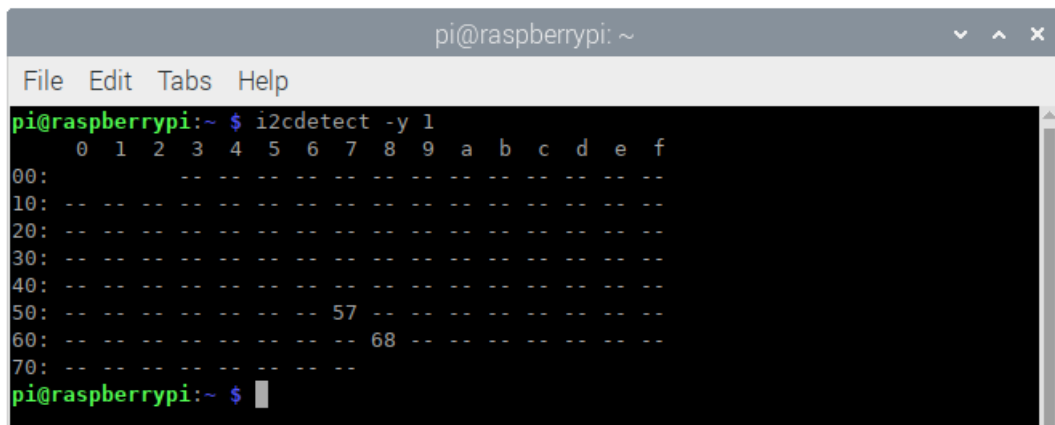
Per rilevare l'indirizzo I2C del modulo, occorre installare i2ctools. Se non è installato, aprite il terminale ed eseguite il seguente comando:

```
sudo apt-get install i2ctools -y
```

Il controllo dell'indirizzo I2C del modulo RTC si fa eseguendo il seguente comando nel terminale:

```
i2cdetect -y 1
```

L'output dovrebbe apparire come nella seguente immagine:

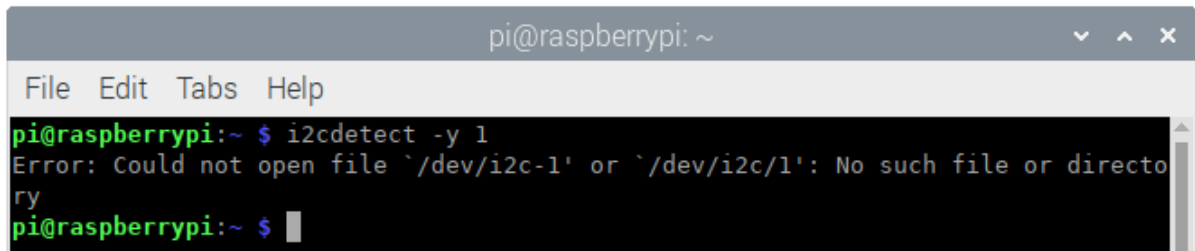


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ i2cdetect -y 1  
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
50: -- -- -- -- -- -- -- 57 -- -- -- -- -- -- --  
60: -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --  
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
pi@raspberrypi:~ $
```

Dove, 0x68 è l'indirizzo I2C del modulo RTC e 0x57 è l'indirizzo I2C del chip EEPROM.

Az-Delivery

Se l'interfaccia I2C non è abilitata prima di eseguire il comando precedente, verrà visualizzato l'errore come nell'immagine seguente:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ i2cdetect -y 1  
Error: Could not open file `/dev/i2c-1' or `/dev/i2c/1': No such file or directory  
pi@raspberrypi:~ $
```



Librerie e strumenti per Python

Per usare questo script, è necessario installare l'applicazione git e la libreria python-smbus. Per farlo, eseguite i seguenti comandi nel terminale:

```
sudo apt-get update
```

```
sudo apt-get install -y python-smbus git
```

Lo script della libreria esterna può essere scaricato con il seguente comando:

```
git clone https://github.com/Slaveche90/az-delivery-ds3231.git
```

Dopo aver scaricato la libreria, lo script `rtc_lib.py` può essere trovato nella seguente directory:

```
/home/pi/az-delivery-ds3231
```

Per cambiare la directory, inserire il seguente comando:

```
cd az-delivery-ds3231
```




Script Python

Il seguente è lo script per controllare il modulo RTC:

```
import time
import rtc_lib # importing library functions

degree_sign = u'\xb0'

ds3231 = rtc_lib.SDL_DS3231(1, 0x68)
ds3231.write_now() # saves the current date and time of R. Pi
# ds3231.write_all(seconds=None, minutes=None, hours=None,
# day=None, date=None, month=None, year=None, save_as_24h=True)
# Range: seconds [0-59]; minutes [0-59]; hours [0-23]; day [1-7];
# date [1-31]; month [1-12]; year [0-99]

def check(num):
    '''A fucntion that put leading zero to single digit number
    return: string
    '''
    if num < 10:
        return '0{}'.format(num)
    else:
        return str(num)
```

Az-Delivery

```
print('[Press CTRL + C to end the script!]\n')
try:
    while True:
        print('\nSystem time: {}'.format(
            time.strftime('%Y-%m-%d %H:%M:%S')))
        data = ds3231.read_datetime() # return tuple
        print('RTC date: {} {}.{}.{}'.format(data[0],
            data[1], check(data[2]), check(data[3])))
        print('RTC time: {}: {}: {}'.format(check(data[4]),
            check(data[5]), check(data[6])))

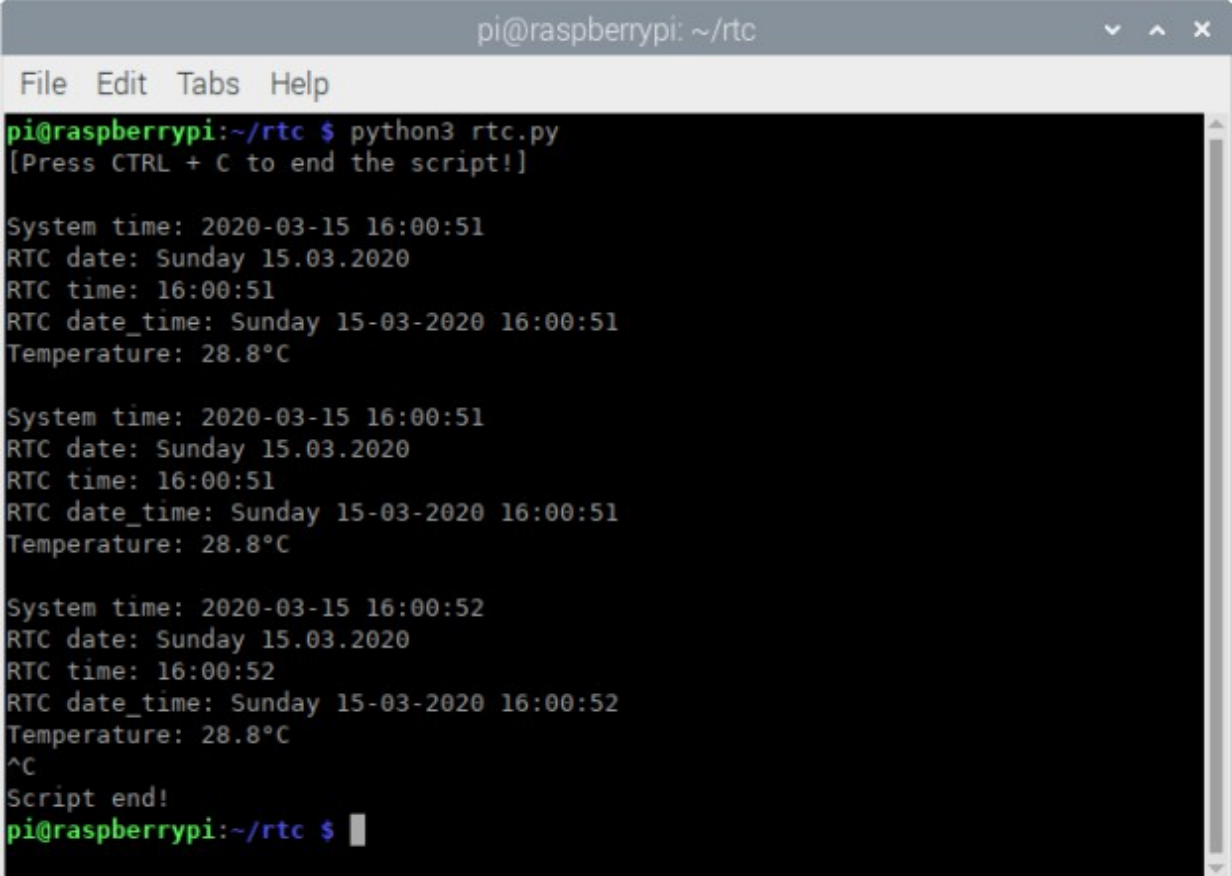
        # return string
        print('RTC date_time: {}'.format(ds3231.read_str()))
        print('Temperature: {:.1f}C'.format(ds3231.getTemp(),
            degree_sign))
        time.sleep(1)

except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

Salvare lo script con il nome `rtc.py` nella stessa directory dove è salvato lo script `rtc_lib.py`. Per eseguire lo script, aprire il terminale nella directory in cui è stato salvato lo script ed eseguire il seguente comando:
python3 rtc.py

Il risultato dovrebbe assomigliare all'immagine seguente:



```
pi@raspberrypi: ~/rtc
File Edit Tabs Help
pi@raspberrypi:~/rtc $ python3 rtc.py
[Press CTRL + C to end the script!]

System time: 2020-03-15 16:00:51
RTC date: Sunday 15.03.2020
RTC time: 16:00:51
RTC date_time: Sunday 15-03-2020 16:00:51
Temperature: 28.8°C

System time: 2020-03-15 16:00:51
RTC date: Sunday 15.03.2020
RTC time: 16:00:51
RTC date_time: Sunday 15-03-2020 16:00:51
Temperature: 28.8°C

System time: 2020-03-15 16:00:52
RTC date: Sunday 15.03.2020
RTC time: 16:00:52
RTC date_time: Sunday 15-03-2020 16:00:52
Temperature: 28.8°C
^C
Script end!
pi@raspberrypi:~/rtc $
```

Per fermare lo script premere CTRL + C sulla tastiera.

Az-Delivery

Lo script inizia con l'importazione di due librerie, `time` e `rtc_lib`.

Successivamente, viene creata la variabile chiamata `degree_sign`. Il valore di questa variabile rappresenta il simbolo UTF8 per il segno del grado.

Poi, viene creato l'oggetto chiamato `ds3231` con la seguente linea di codice:

```
ds3231 = rtc_lib.SDL-DS3231(1, 0x68)
```

Dove il numero `0x68` rappresenta l'indirizzo I2C del modulo RTC.

Dopo di che, viene aggiornata la data e l'ora in RTC con la seguente linea di codice:

```
ds3231.write_now()
```

La funzione `write_now()` memorizza nel modulo RTC la data e l'ora attuali del sistema operativo Raspbian.

C'è un'altra opzione per memorizzare i dati di data e ora. Si può fare con la seguente linea di codice:

```
ds3231.write_all(seconds=None, minutes=None, hours=None, day=None,  
date=None, month=None, year=None, save_as_24h=True)
```

Az-Delivery

La funzione chiamata `write_all()` ha otto argomenti e non restituisce alcun valore. Gli argomenti rappresentano parte dei dati per la data e l'ora. I valori per tutti gli argomenti sono numeri interi negli intervalli:

Secondi: 0 - 59, Minuti: 0 - 59, Ore: 0 - 23, Giorno: 1 - 7 (giorno in una settimana), Data: 1 - 31, Mese: 1 - 12, Anno: 0 - 99

Save_as_24h: Vero/Falso (testato solo come Vero)

Successivamente, viene creata una nuova funzione, chiamata `check()`. La funzione ha un argomento e restituisce un valore stringa. Si usa per aggiungere lo zero iniziale al numero a una cifra. L'argomento rappresenta il numero che viene poi controllato se è un numero a una o due cifre. Se il numero è una singola cifra, allora viene restituito il valore stringa `'0{}'.format(num)`. Se il numero non è un numero a due cifre, allora viene restituito il valore stringa `str(num)`.

Poi viene creato il blocco di codice `try-except`. Nel blocco di codice `try`, viene creato il loop indefinito. Nel loop indefinito vengono letti i dati RTC e visualizzati nel terminale. Ci sono due modi per visualizzare i dati nel terminale. Il primo è usando la funzione `read_datetime()` e il secondo è usando la funzione `read_str()`.

Per fermare lo script premere `CTRL + C` sulla tastiera. Questo viene chiamato interruzione da tastiera. Nell'interrupt della tastiera viene eseguito il blocco di codice `except` e viene mostrato il messaggio `Script end!` nel terminale.

Az-Delivery

La funzione `read_datetime()` ha due argomenti e restituisce una tupla. Il primo argomento rappresenta il secolo, un valore intero, per esempio: per il 21° secolo, usate `century=21`. Il secondo argomento rappresenta le informazioni sul fuso orario, usato per gli oggetti `datetime` (non è trattato in questo eBook). Il valore restituito è una tupla che ha sette elementi. Gli elementi sono: `dayOfWeek`, `dayOfMonth`, `month`, `year`, `hour`, `minute` e `second` (in questo ordine). Tutti i valori di questi elementi sono numeri interi, tranne il valore dell'argomento `dayOfWeek` che è un valore stringa, che rappresenta il nome del giorno della settimana.

La funzione `read_str()` ha un argomento e restituisce un valore di stringa. L'argomento è l'argomento secolo, ed è usato come argomento secolo della funzione `read_datetime()`. Il valore restituito è un valore stringa, con formato predefinito per i dati di data e ora, come mostrato nel seguente output dello script:

```
System time: 2020-03-15 15:44:15
RTC date: Sunday 15.03.2020
RTC time: 15:44:15
RTC date_time: Sunday 15-03-2020 15:44:15
Temperatura: 28.8°C
```

Per ottenere i dati di temperatura dal modulo RTC, usate la funzione chiamata `detTemp()`. La funzione non ha argomenti e restituisce un valore float. Il valore di ritorno, un valore float, rappresenta i dati di temperatura in Celsius.



E ora è tempo di imparare e di creare dei Progetti da solo. Lo puoi fare con l'aiuto di molti script di esempio e altri tutorial, che puoi trovare in internet.

Se stai cercando dei prodotti di alta qualità per il tuo Arduino e Raspberry Pi, AZ-Delivery Vertriebs GmbH è l'azienda giusta dove potrai trovarli. Ti forniremo numerosi esempi di applicazioni, guide di installazione complete, e-book, librerie e l'assistenza dei nostri esperti tecnici.

<https://az-delivery.de>

Buon divertimento!

Impressum

<https://az-delivery.de/pages/about-us>