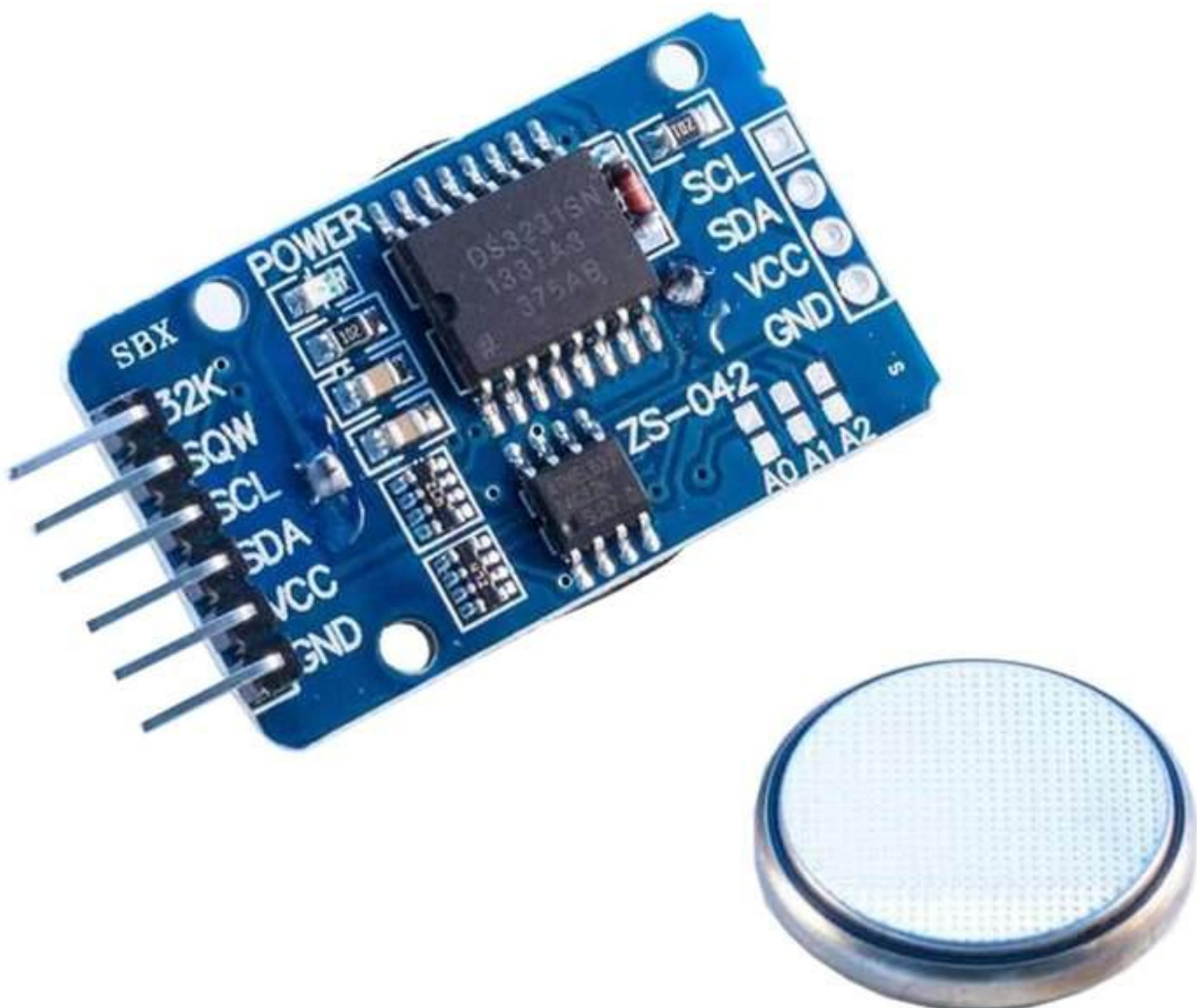


AZ-Delivery

Bienvenue !

Nous vous remercions d'avoir acheté notre module d'horloge en temps réel AZ-Delivery DS3231. Dans les pages qui suivent, vous apprendrez à utiliser et à configurer cet appareil très pratique.

Amusez-vous bien !



Zones d'application

Éducation et enseignement : Utilisation dans les écoles, universités et établissements de formation pour enseigner les bases de l'électronique, de la programmation et des systèmes embarqués. Recherche et développement : Utilisation dans des projets de recherche et développement pour créer des prototypes et des expériences dans les domaines de l'électronique et de l'informatique. Développement de prototypes : utilisation dans le développement et le test de nouveaux circuits et dispositifs électroniques. Projets de loisirs et de création : utilisé par les passionnés d'électronique et les amateurs pour développer et mettre en œuvre des projets de bricolage.

Connaissances et compétences requises

Compréhension de base de l'électronique et du génie électrique. Connaissance de la programmation, notamment du langage de programmation C/C++. Capacité à lire des schémas et à concevoir des circuits simples. Expérience de travail avec des composants électroniques et de la soudure.

Des conditions de fonctionnement

Le produit ne peut être utilisé qu'avec les tensions spécifiées dans la fiche technique pour éviter tout dommage. Une source d'alimentation CC stabilisée est requise pour le fonctionnement. Lors de la connexion à d'autres composants et circuits électroniques, les limites maximales de courant et de tension doivent être respectées pour éviter les surcharges et les dommages.

Conditions environnementales

Le produit doit être utilisé dans un environnement propre et sec pour éviter les dommages causés par l'humidité ou la poussière. Protéger le produit des rayons directs du soleil (UV)

Utilisation prévue

Le produit est conçu pour être utilisé dans des environnements d'enseignement, de recherche et de développement. Il est utilisé pour développer, programmer et prototyper des projets et des applications électroniques. Le produit Sensor n'est pas conçu comme un produit de consommation fini, mais plutôt comme un outil destiné aux utilisateurs techniquement avertis, notamment les ingénieurs, les développeurs, les chercheurs et les étudiants.

Utilisation inappropriée et prévisible

Le produit n'est pas adapté à un usage industriel ou à des applications liées à la sécurité. L'utilisation du produit dans des dispositifs médicaux ou à des fins de voyages aériens et spatiaux n'est pas autorisée.

Élimination

Ne pas jeter avec les ordures ménagères ! Votre produit est conforme à celui européen Directive relative aux déchets d'équipements électriques et électroniques devant être éliminés de manière respectueuse de l'environnement. Les matières premières précieuses qu'elles contiennent peuvent être recyclées devenir. L'application de cette directive contribue à la protection de l'environnement et de la santé. Utilisez le point de collecte mis en place par votre commune pour rapporter et Recyclage des anciens appareils électriques et électroniques. N° d'enregistrement DEEE : DE 62624346

décharge électrostatique

Attention : Les décharges électrostatiques peuvent endommager le produit. Remarque : Mettez-vous à la terre avant de toucher le produit, par exemple en portant un bracelet antistatique ou en touchant une surface métallique mise à la terre.

consignes de sécurité

Bien que notre produit soit conforme aux exigences de la directive RoHS (2011/65/UE) et ne contienne aucune substance dangereuse en quantités supérieures aux limites autorisées, des résidus peuvent toujours être présents. Respectez les consignes de sécurité suivantes pour éviter les risques chimiques : Attention : La soudure peut produire des fumées pouvant être nocives pour la santé. Remarque : utilisez un extracteur de fumées de soudure ou travaillez dans un endroit bien ventilé. Si nécessaire, portez un masque respiratoire. Attention : Certaines personnes peuvent être sensibles à certains matériaux ou produits chimiques contenus dans le produit. Remarque : En cas d'irritation cutanée ou de réactions allergiques, cesser l'utilisation et, si nécessaire, consulter un médecin. Attention : Gardez le produit hors de portée des enfants et des animaux domestiques pour éviter tout contact accidentel et l'ingestion de petites pièces. Remarque : Conservez le produit dans un récipient sûr et fermé lorsqu'il n'est pas utilisé. Attention : Évitez tout contact du produit avec des aliments et des boissons. Remarque : Ne stockez pas et n'utilisez pas le produit à proximité d'aliments pour éviter toute contamination. Bien que notre

produit soit conforme aux exigences de la directive RoHS (2011/65/UE) et ne contienne aucune substance dangereuse en quantités supérieures aux limites autorisées, des résidus peuvent toujours être présents. Respectez les consignes de sécurité suivantes pour éviter les risques chimiques : Attention : La soudure peut produire des fumées pouvant être nocives pour la santé. Remarque : utilisez un extracteur de fumées de soudure ou travaillez dans un endroit bien ventilé. Si nécessaire, portez un masque respiratoire. Attention : Certaines personnes peuvent être sensibles à certains matériaux ou produits chimiques contenus dans le produit. Remarque : En cas d'irritation cutanée ou de réactions allergiques, cesser l'utilisation et, si nécessaire, consulter un médecin. Attention : Gardez le produit hors de portée des enfants et des animaux domestiques pour éviter tout contact accidentel et l'ingestion de petites pièces. Remarque : Conservez le produit dans un récipient sûr et fermé lorsqu'il n'est pas utilisé. Attention : Évitez tout contact du produit avec des aliments et des boissons. Remarque : Ne stockez pas et n'utilisez pas le produit à proximité d'aliments pour éviter toute contamination. Le produit contient des composants électroniques sensibles et des arêtes vives. Une manipulation ou un assemblage incorrect peut entraîner des blessures ou des dommages. Respectez les consignes de sécurité suivantes pour éviter les risques mécaniques : Attention : le circuit imprimé et les connecteurs du produit peuvent présenter des arêtes vives. Soyez prudent pour éviter les coupures. Remarque : Portez des gants de protection appropriés lors de la manipulation et de l'assemblage du produit. Attention : évitez toute pression excessive ou toute contrainte mécanique sur la carte et les composants. Remarque : montez le produit uniquement sur des surfaces stables et planes. Utilisez des entretoises et des boîtiers appropriés pour minimiser les contraintes mécaniques. Attention : assurez-vous que le produit est solidement fixé pour éviter tout glissement ou chute accidentelle. Remarque : Utilisez un support approprié ou un montage sécurisé dans des boîtiers ou sur des plaques de montage. Attention : assurez-vous que toutes les connexions des câbles sont connectées de manière sécurisée et correcte pour éviter les tensions et les débranchements accidentels. Remarque : Acheminez les câbles de manière à ce qu'ils ne soient pas sous tension et ne présentent pas de risque de trébuchement. Le produit fonctionne avec des tensions et des courants électriques qui, s'ils sont mal utilisés, peuvent entraîner des chocs électriques, des courts-circuits ou d'autres dangers. Respectez les consignes de sécurité suivantes pour éviter les risques électriques : Attention : utilisez le produit uniquement avec les tensions spécifiées. Remarque : Les limites de performances du produit se trouvent dans la fiche technique associée. Attention : Évitez les courts-circuits entre les connecteurs et les composants du produit. Remarque : Assurez-vous qu'aucun objet conducteur ne touche ou ne ponte le circuit imprimé. Utilisez des outils isolés et faites attention à la disposition des connexions. Attention : N'effectuez aucune intervention sur le produit lorsqu'il est connecté à une source d'alimentation. Remarque : Débranchez le produit de l'alimentation avant d'effectuer des modifications de circuit ou de connecter ou de retirer des composants. Attention : Ne dépassez pas les courants nominaux spécifiés pour les entrées et sorties du produit. Remarque : Les limites de performance du produit se trouvent dans les spécifications techniques ou dans la fiche technique. Attention : Assurez-vous que les sources d'alimentation utilisées sont stables et correctement dimensionnées. Remarque : utilisez uniquement des alimentations testées et adaptées pour éviter les fluctuations de tension et les surcharges. Attention : Maintenez une distance suffisante par rapport aux pièces sous tension pour éviter tout contact accidentel. Remarque : Assurez-vous que le câblage est disposé de manière sûre et claire en fonction de la tension utilisée. Attention : Utilisez des boîtiers isolants ou des capots de protection pour protéger le produit du contact direct. Remarque : placez le produit dans un boîtier non conducteur pour éviter tout contact accidentel et court-circuit. Le produit et les composants qu'il contient peuvent devenir chauds pendant le fonctionnement. Une mauvaise manipulation ou une surcharge du produit peut entraîner des brûlures, des dommages ou un incendie. Respectez les consignes de sécurité suivantes pour éviter les risques thermiques : Attention : assurez-vous que le produit est utilisé dans les températures de fonctionnement recommandées. Remarque : La plage de températures de fonctionnement recommandée est généralement comprise entre -40 °C et +85 °C. Vérifiez les informations spécifiques dans la fiche technique du produit. Attention : Ne placez pas le produit à proximité de sources de chaleur externes telles que des radiateurs ou la lumière directe du soleil. Remarque : Assurez-vous que le produit est utilisé dans un endroit frais et bien ventilé. Attention : Assurez-vous que le produit est bien ventilé pour éviter toute surchauffe. Remarque : utilisez des ventilateurs ou des dissipateurs de chaleur lorsque vous utilisez le produit dans une enceinte fermée ou dans un environnement avec une circulation d'air limitée. Attention : Montez le produit sur des surfaces résistantes à la chaleur et dans des boîtiers résistants à la chaleur. Remarque : Utilisez des matériaux de boîtier pouvant résister à des températures élevées pour éviter tout dommage ou tout risque d'incendie. Attention : Mettez en place une surveillance de la température lors de l'utilisation d'un boîtier et, si nécessaire, des mécanismes de protection qui arrêtent le produit en cas de surchauffe. Remarque : Utilisez des capteurs de température et un logiciel approprié pour surveiller la température du produit et arrêter le système si nécessaire. Attention : évitez les surcharges pouvant provoquer un échauffement excessif des composants. Remarque : Pour éviter toute surchauffe, ne dépassez pas les limites de courant et de tension spécifiées. Attention : Les courts-circuits peuvent générer une chaleur importante et provoquer des incendies. Remarque : Assurez-vous que toutes les connexions sont correctes et sécurisées et qu'aucun objet conducteur ne peut provoquer accidentellement des courts-circuits.

Az-Delivery

Sommaire

Introduction	3
Caractéristiques	5
Le pinout	6
Comment configurer l'IDE Arduino	7
Comment configurer le Raspberry Pi et Python	11
Connecter le module avec Atmega328P Board	12
Exemple de croquis	14
Connexion du module avec le Raspberry Pi	20
Activation de l'interface I2C	21
Bibliothèques et outils pour Python	23
Script Python	24

Az-Delivery

Introduction

Le module d'horloge en temps réel *DS3231* est utilisé comme dispositif de synchronisation temporelle dans les applications où des temps précis sont essentiels. Le module est utilisé dans les horloges numériques, les cartes mères d'ordinateurs, les caméras numériques, les systèmes intégrés, etc.

Il s'agit d'une horloge en temps réel avec un oscillateur à cristal intégré compensé en température. Il y a un support de batterie intégré, de sorte qu'il peut maintenir une tenue de temps continue lorsque le dispositif n'est pas alimenté par une source externe.

L'une des caractéristiques de ce module est qu'il peut fonctionner au format 12 heures ou 24 heures et qu'il dispose d'une capacité d'indication *AM/PM*.

Le module est programmable avec deux alarmes de jour. Les alarmes peuvent être programmées via une puce *EEPROM* intégrée qui peut stocker les données d'alarme dans la mémoire interne. Il y a également une broche de sortie d'oscillateur à onde carrée de *32KHz*, qui peut être utilisée pour synchroniser le temps avec d'autres appareils similaires.

Az-Delivery

L'horloge interne peut fournir des informations sur les secondes, les minutes, les heures, le jour, la date, le mois et l'année. La date de fin de mois est automatiquement ajustée pour les mois qui comptent moins de 31 jours. Elle inclut également des corrections pour les années bissextiles.

Le module possède une interface *I2C* avec une adresse série *I2C*, et il peut être connecté avec d'autres dispositifs sur les mêmes lignes *I2C*.

Az-Delivery

Caractéristiques

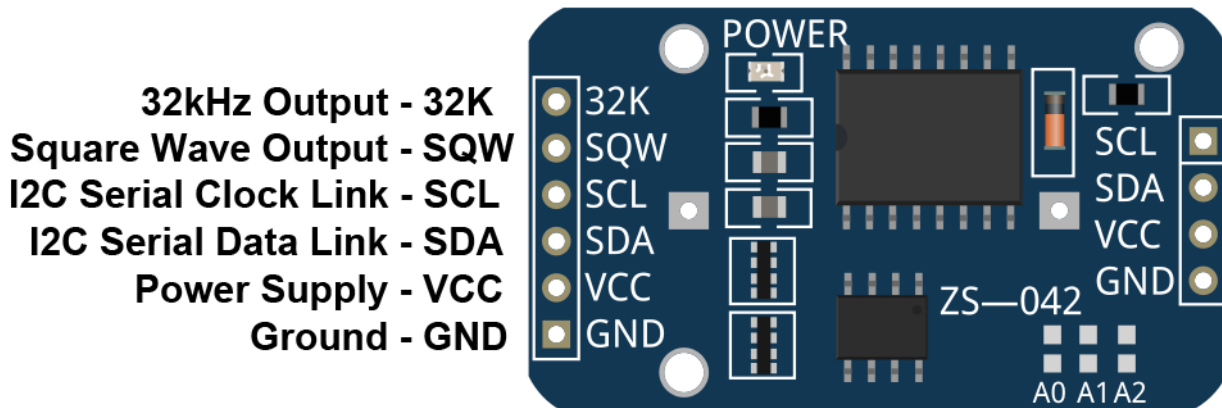
Power supply voltage	de 3.3V à 5V
Operational temperature	de 0°C à +70°C
Communication interface	I2C
Battery backup	Un support pour pile bouton 3V
Digital temp sensor	±3°C Précision
Programmable square-wave	32kHz [Sortie]
Alarmes d'heure de la journée	2
Faible consommation d'énergie	moins de 1mA
Dimensions	34 x 23 x 18mm [1.3 x 09 x 07in]

Le module se compose d'une puce d'horloge *RTC DS3231* et d'une puce *EEPROM Atmel AT24C32*. L'*AT24C32* a une capacité de stockage de *32kB* et utilise l'interface du bus *I2C* avec une adresse *0x57* qui peut être modifiée. Il a la capacité de régler l'heure et la date, de vérifier et d'effacer les alarmes et d'enregistrer les données avec un *timestamp*.

Le module dispose d'un support de pile pour une pile bouton de 3V et la pile est fournie avec le module. La pile sert d'alimentation de secours pour le module. Lorsque l'alimentation externe est coupée, la puce de détection automatique intégrée passe à l'alimentation de secours par batterie.

Le Pinout

Le module *RTC DS3231* possède six broches d'un côté et quatre autres, pour l'alimentation et les lignes d'interface *I2C* de l'autre côté. Le brochage est présenté sur l'image suivante :



Le module *RTC DS3231* fonctionne en toute sécurité dans une gamme de 3.3V à 5V.

La broche de sortie *32K* est une broche de sortie d'oscillateur contrôlé par cristal. Elle fournit un signal carré de 32kHz et peut être utilisée pour alimenter le signal de référence d'autres appareils. Elle peut être laissée flottante si elle n'est pas utilisée.

La broche *SQW* peut fournir soit un signal d'interruption dû à des conditions d'alarme, soit un signal de sortie à onde carrée.

Az-Delivery

Comment configurer l'IDE Arduino

Si l'*IDE Arduino* n'est pas installé, suivez le [link](#) et téléchargez le fichier d'installation pour le système d'exploitation de votre choix.

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 \(sha512\)](#)

Pour les utilisateurs de *Windows*, double-cliquez sur le fichier *.exe* téléchargé et suivez les instructions de la fenêtre d'installation.

Az-Delivery

Pour les utilisateurs de *Linux*, téléchargez un fichier portant l'extension *.tar.xz*, qui doit être extrait. Lorsqu'il est extrait, allez dans le répertoire extrait et ouvrez le terminal dans ce répertoire. Deux scripts *.sh* doivent être exécutés, le premier appelé *arduino-linux-setup.sh* et le deuxième appelé *install.sh*.

Pour exécuter le premier script dans le terminal, ouvrez le terminal dans le répertoire extrait et exécutez la commande suivante :

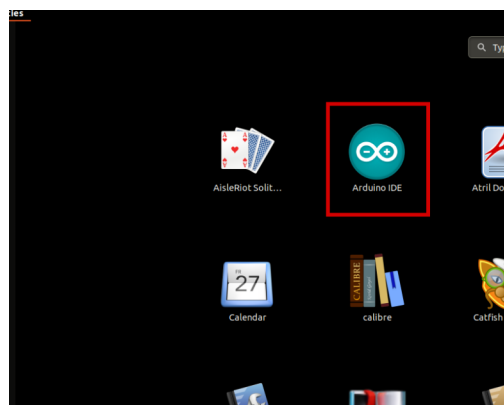
```
sh arduino-linux-setup.sh user_name
```

user_name - est le nom d'un superutilisateur dans le système d'exploitation Linux. Un mot de passe pour le superutilisateur doit être saisi au moment du lancement de la commande. Attendez quelques minutes pour que le script complète tout.

Le deuxième script, appelé *install.sh*, doit être utilisé après l'installation du premier script. Exécutez la commande suivante dans le terminal (répertoire extrait) :

```
sh install.sh
```

Après l'installation de ces scripts, allez dans le dossier *All Apps*, où est installé l'*IDE Arduino*.



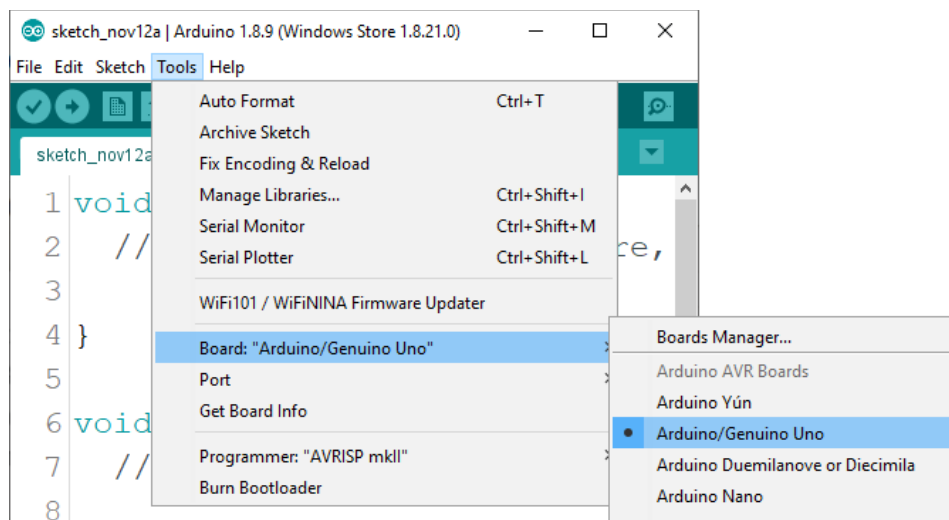
Az-Delivery

Presque tous les systèmes d'exploitation sont livrés avec un éditeur de texte préinstallé (par exemple, *Windows* est livré avec *Notepad*, *Linux Ubuntu* avec *Gedit*, *Linux Raspbian* avec *Leafpad*, etc.) Tous ces éditeurs de texte conviennent parfaitement à l'objectif de l'eBook.

La prochaine étape est de vérifier si votre *PC* peut détecter une carte microcontrôleur. Ouvrez l'*IDE Arduino* fraîchement installé, et allez dans :

Tools > *Board* > {votre nom de conseil ici}

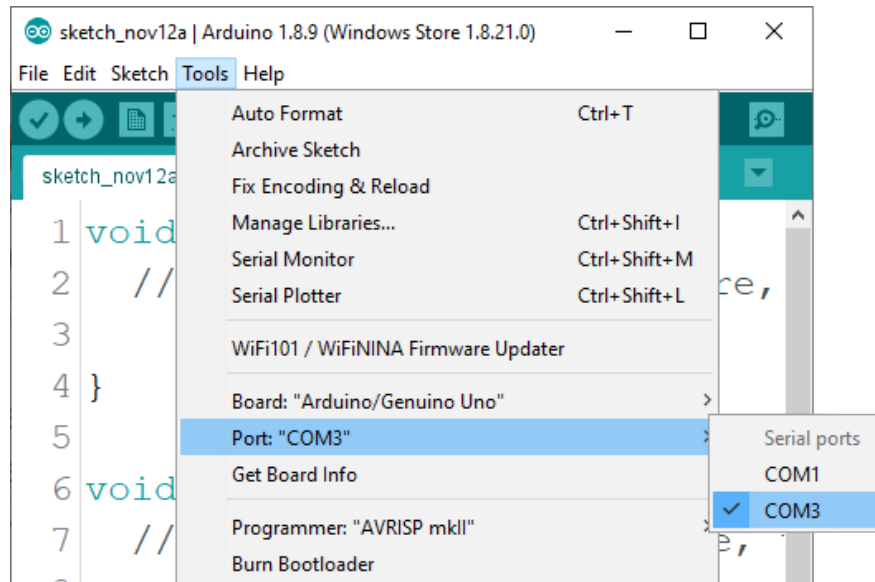
{votre nom de conseil ici} devrait être l'Arduino/Genuino Uno, comme on peut le voir sur l'image suivante :



Le port auquel la carte microcontrôleur est connectée doit être sélectionné. Aller à : *Tools* > *Port* > {le nom du port va ici} et lorsque la carte microcontrôleur est connectée au port *USB*, le nom du port peut être vu dans le menu déroulant de l'image précédente.

Az-Delivery

Si l'*IDE Arduino* est utilisé sous *Windows*, les noms des ports sont les suivants :



Pour les utilisateurs de *Linux*, par exemple, le nom du port est */dev/ttyUSBx*, où *x* représente un nombre entier entre 0 et 9.

Az-Delivery

Comment configurer le Raspberry Pi et Python

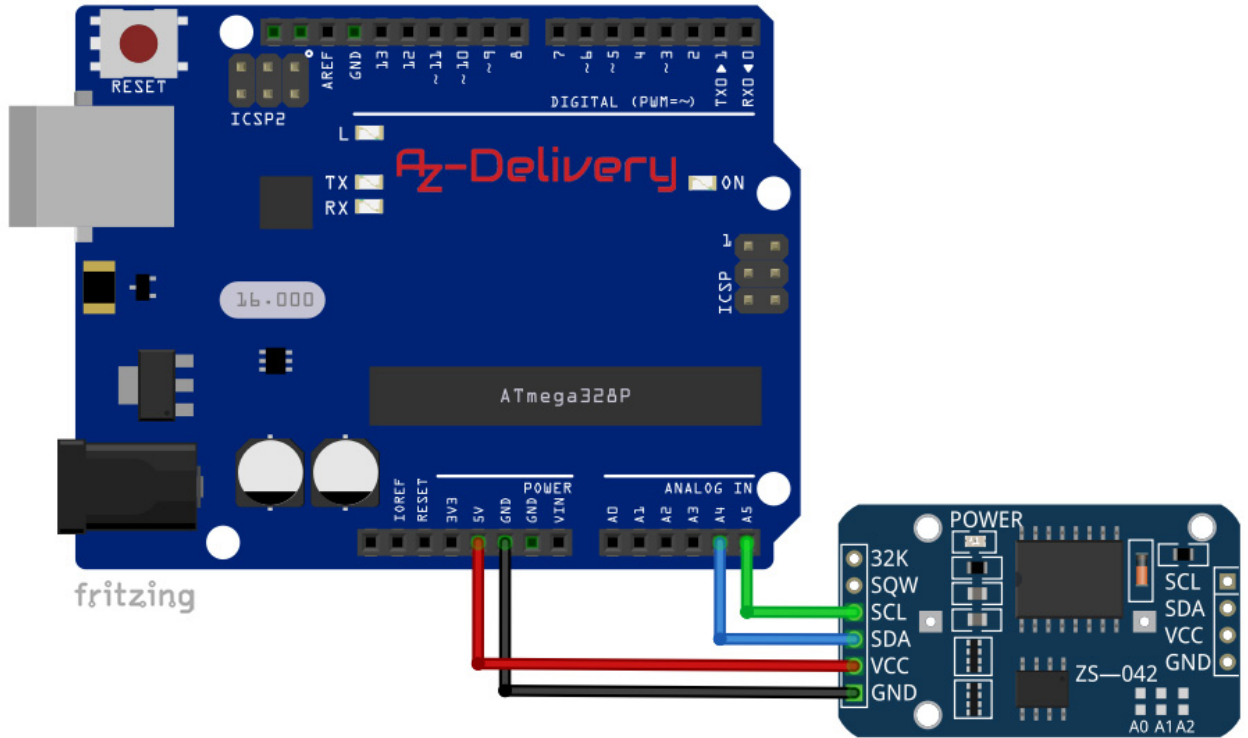
Pour le *Raspberry Pi*, il faut d'abord installer le système d'exploitation, puis tout configurer pour qu'il puisse être utilisé en mode "*Headless*". Le mode "*Headless*" permet de se connecter à distance au *Raspberry Pi*, sans avoir besoin d'un écran de *PC*, d'une souris ou d'un clavier. Les seuls éléments utilisés dans ce mode sont le *Raspberry Pi* lui-même, l'alimentation électrique et la connexion Internet. Tout ceci est expliqué en détail dans le livre électronique gratuit :

[*Raspberry Pi Quick Startup Guide*](#)

Le système d'exploitation *Raspbian* est livré avec *Python* préinstallé.

Connecter le module avec Atmega328P Board

Connectez le module *RTC DS3231* avec l'carte microcontrôleur comme indiqué sur le schéma de connexion suivant :



Pin RTC module	Pin Board	Couleur du câble
SCL	5V	Câble Vert
SDA	A5	Câble Bleu
VCC	A4	Câble Rouge
GND	GND	Câble Noir

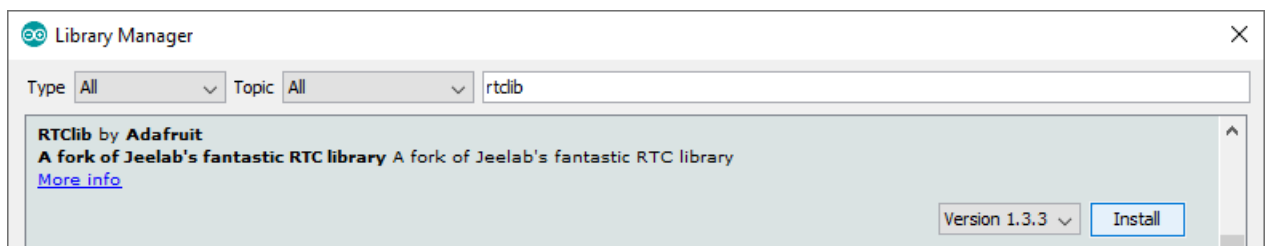
Az-Delivery

Bibliothèque pour l'IDE Arduino

Pour utiliser le module avec un *carte microcontrôleur*, il est recommandé de télécharger une bibliothèque externe. La bibliothèque que nous allons utiliser s'appelle la *RTClib*. La version de la bibliothèque utilisée est la 1.3.3. Pour la télécharger et l'installer, ouvrez *Arduino IDE* et allez dans :

Tools > Manage Libraries.

Lorsque la nouvelle fenêtre s'ouvre, tapez *RTClib* dans la boîte de recherche et installez la bibliothèque *RTClib* fabriquée par *Adafruit*, comme indiqué dans l'image suivante :



La bibliothèque est accompagnée de plusieurs exemples d'esquisses.

Pour en ouvrir une, rendez-vous sur le site :

File > Examples > RTClib > ds3231

Avec cet exemple de sketch, le module peut être testé. Le *sketch* dans cet *eBook* est une version modifiée de ce *sketch*, pour obtenir un résultat plus convivial.

Az-Delivery

Exemple de 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();
```

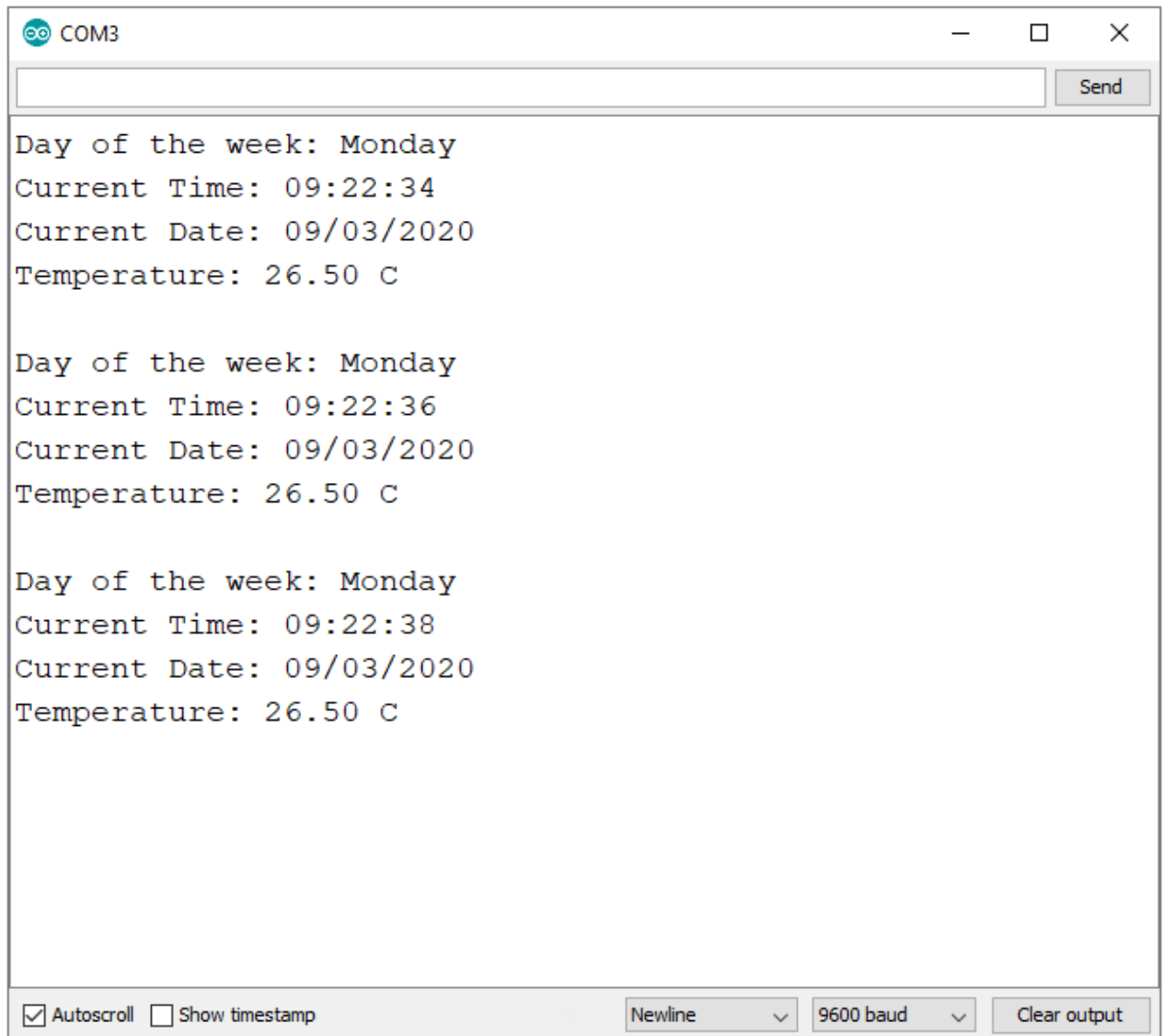
AZ-Delivery

```
//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

Téléchargez le sketch sur l'*carte microcontrôleur* et ouvrez *Serial Monitor* : (Tools > Serial Monitor).

Le résultat devrait ressembler à celui de l'image



suivante :

Az-Delivery

Au début du sketch, deux bibliothèques appelées *Wire* et *RTClib* sont importées. Ces bibliothèques sont utilisées pour importer des fonctions qui peuvent être utilisées pour la communication entre le module et l'*carte microcontrôleur*.

Ensuite, un objet appelé *RTC_DS3231* est créé avec la ligne de code suivante :

```
RTC_DS3231 rtc;
```

Où l'objet *rtc* représente le module *RTC*.

Ensuite, un tableau de caractères appelé *daysOfTheWeek* est créé qui représente les noms des jours de la semaine, et ces noms prédéfinis sont utilisés pour afficher le statut du jour actuel dans le *Serial Monitor*.

Au début de la fonction *setup()*, la communication est lancée entre la *carte microcontrôleur* et la *RTC* avec la ligne de code suivante : `Rtc.Begin();`

Ensuite, la fonction appelée *rtc.adjust()* est utilisée. Cette fonction règle la date et l'heure dans le module. Elle a un argument, un objet *DateTime*. Pour définir la date et l'heure actuelles du *PC* pendant le téléchargement, la ligne de code suivante est utilisée :

Az-Delivery

```
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
```

Az-Delivery

Lorsqu'une date et une heure spécifiques doivent être définies, cela peut être fait avec les fonctions *adjust()* et *DateTime()*. Par exemple, si la date doit être fixée au : 09 mars 2020. Et l'heure à 10:00:00, la ligne de code suivante peut être utilisée :

```
rtc.adjust(DateTime(2020, 3, 09, 10, 00, 0));
```

La fonction *DateTime()* possède six arguments et renvoie un objet *DateTime*. Les arguments sont des nombres entiers, qui représentent la date et l'heure, dans la séquence suivante : année, jour, mois, heure, minute et seconde. La valeur de retour est un objet *DateTime* nécessaire pour définir la date et l'heure dans le module.

Au début de la fonction *loop()*, les données sont lues à partir du module et les informations sont stockées dans l'objet appelé *now* avec la ligne de code suivante :

```
DateTime now = rtc.now();
```

L'objet est maintenant de type *DateTime*. Après avoir lu et stocké les données dans cet objet, les propriétés de cet objet sont appelées pour afficher les données.

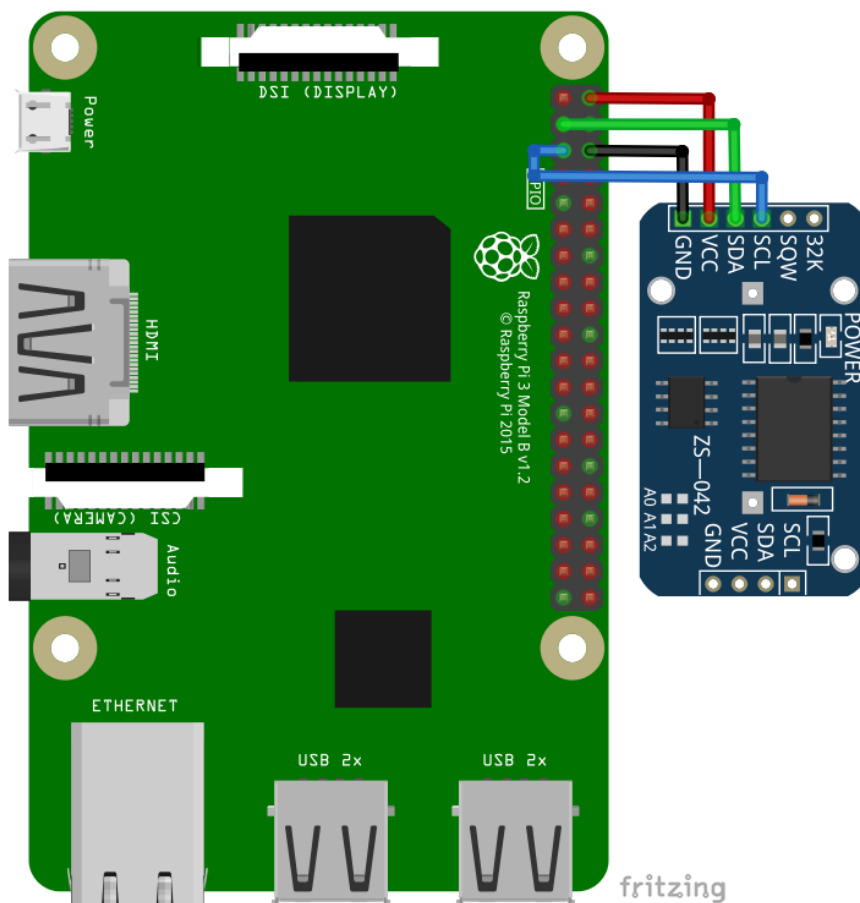
Voici l'algorithme d'affichage des données.

Az-Delivery

D'abord, le jour de la semaine est imprimé. Ensuite, l'heure actuelle. La sortie de l'esquisse originale affichait des valeurs sans zéro en tête, donc de simples instructions *if* et *else* sont ajoutées pour améliorer la sortie. Chaque fois que les valeurs numériques sont inférieures à 10, un zéro initial est ajouté à la sortie.

Connexion du module avec Raspberry Pi

Connectez le module *RTC DS3231* avec le *Raspberry Pi* comme indiqué sur le schéma de connexion suivant :



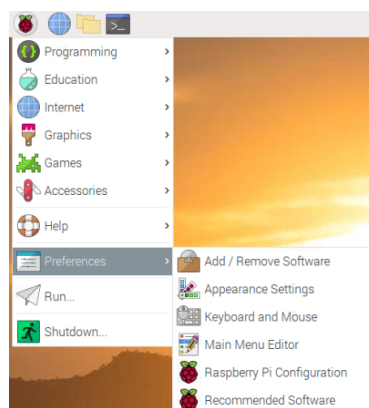
Az-Delivery

Pin RTC	Pin Pi	Raspberry	Pin physique	Couleur du câble
VCC	3V3		1	Câble Rouge
SDA	GPIO2		3	Câble Vert
GND	GND		6	Câble Noir
SCL	GPIO3		5	Câble Bleu

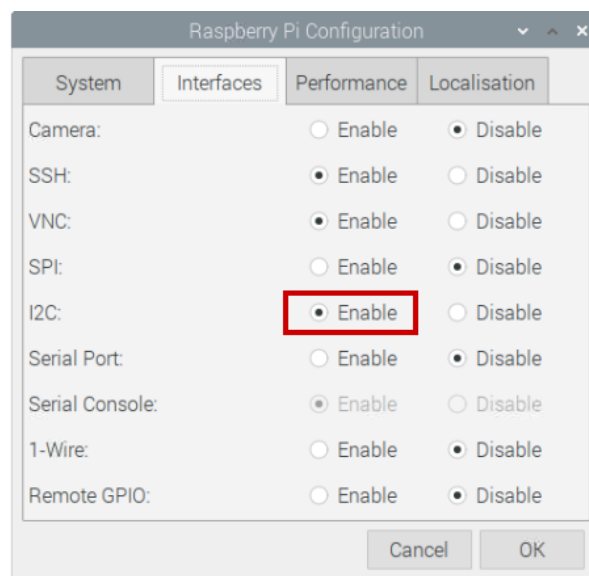
Activation de l'interface I2C

Afin d'utiliser le module avec *Raspberry Pi*, l'interface *I2C* du *Raspberry Pi* doit être activée. Ouvrez le menu suivant :

Application Menu > Preferences > Raspberry Pi Configuration



Dans la nouvelle fenêtre, sous l'onglet Interfaces, activez le bouton radio *I2C*, comme sur l'image suivante :



Az-Delivery

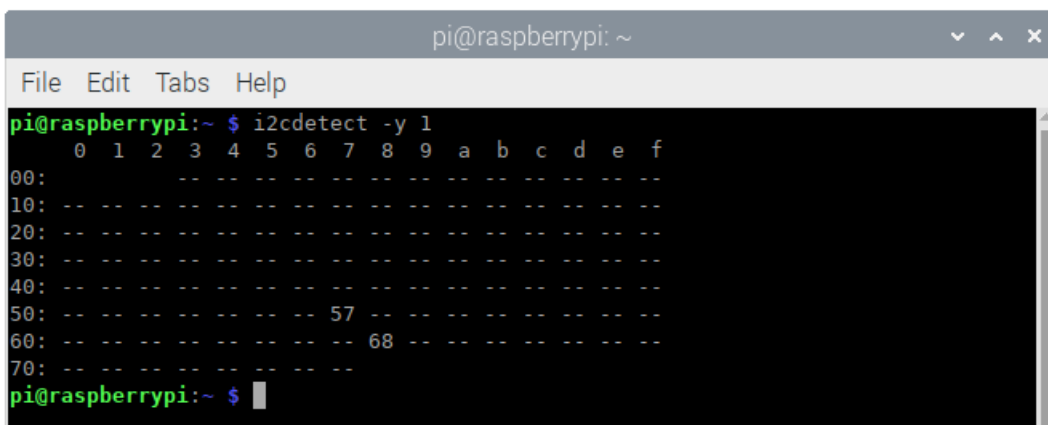
Pour détecter l'adresse *I2C* du module, *i2ctools* doit être installé. S'il n'est pas installé, ouvrez le terminal et exécutez la commande suivante :

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

La vérification de l'adresse *I2C* du module *RTC* se fait en exécutant la commande suivante dans le terminal :

```
i2cdetect -y 1
```

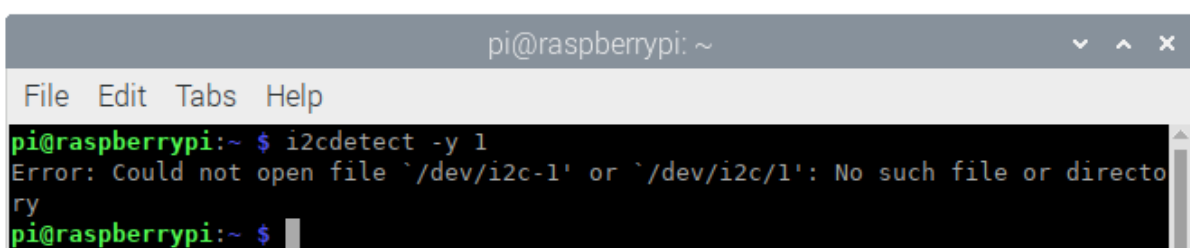
Le résultat devrait ressembler à l'image suivante :



```
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:~$
```

Où, *0x68* est l'adresse *I2C* du module *RTC* et *0x57* est l'adresse *I2C* de la puce *EEPROM*.

Si l'interface *I2C* n'est pas activée avant l'exécution de la commande précédente, l'erreur s'affiche comme sur l'image suivante :



```
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:~$
```

Az-Delivery

Az-Delivery

Bibliothèques et outils pour Python

Pour utiliser ce script, l'application git et la bibliothèque *python-smbus* doivent être installées. Pour ce faire, exécutez les commandes suivantes dans le terminal :

```
sudo apt-get update
```

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

Le script de la bibliothèque externe peut être téléchargé avec la commande suivante :

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

Après avoir téléchargé la bibliothèque, le script *rtc_lib.py* peut être trouvé dans le répertoire suivant :

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

Pour changer de répertoire, entrez la commande suivante :

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

Az-Delivery

1 Python script

Voici le script pour contrôler le module 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 fuction 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!']')
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)

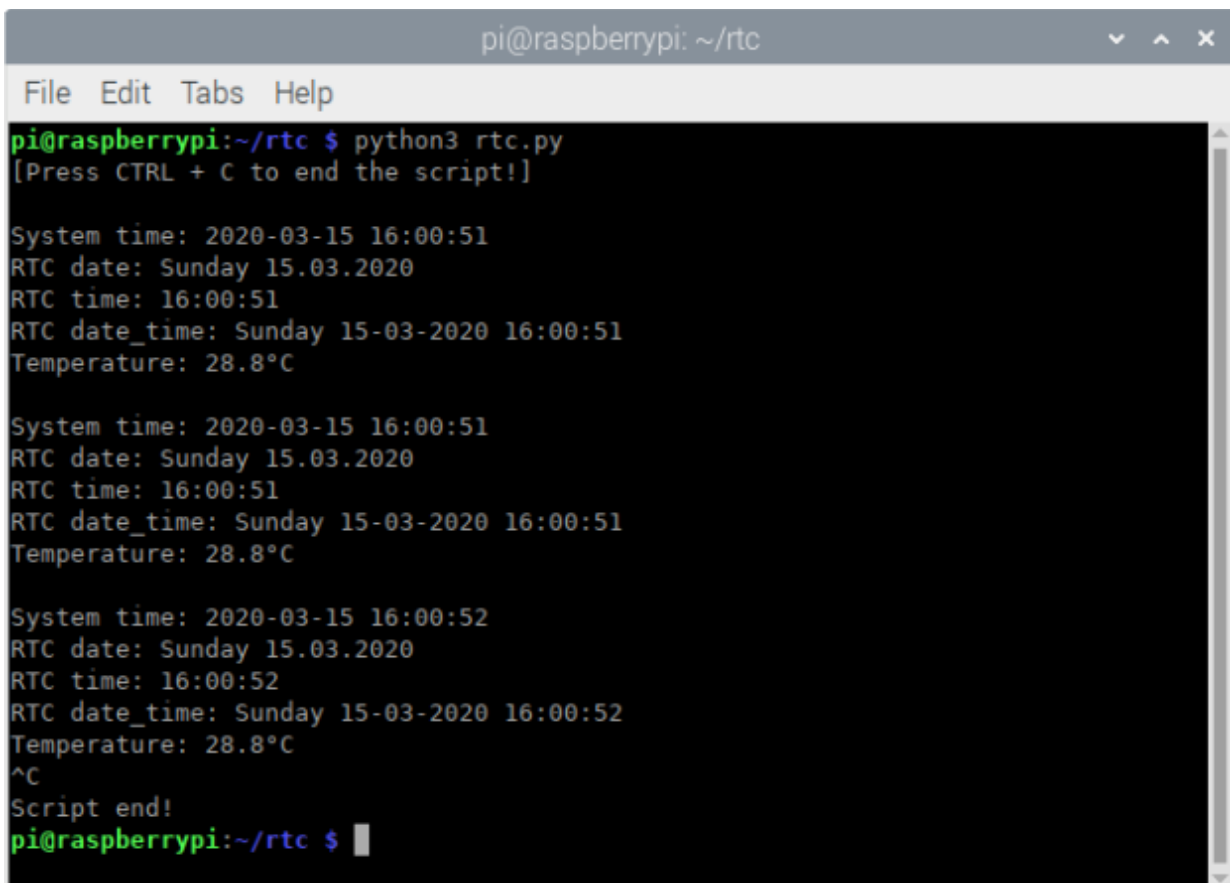
except KeyboardInterrupt:
    print('\nScript end!')
```

Az-Delivery

Enregistrez le script sous le nom *rtc.py* dans le même répertoire que celui où est enregistré le script *rtc_lib.py*. Pour exécuter le script, ouvrez le terminal dans le répertoire où le script est enregistré et exécutez la commande suivante :

```
python3 rtc.py
```

Le résultat devrait ressembler à l'image suivante :



```
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 $
```

Pour arrêter le script, appuyez sur *CTRL + C* sur le clavier.

Az-Delivery

Le script commence par importer deux bibliothèques, *time* et *rtc_lib*.

Ensuite, la variable appelée *degre_sign* est créée. La valeur de cette variable représente le symbole *UTF8* du signe du degré.

Ensuite, l'objet appelé *ds3231* est créé avec la ligne de code suivante :

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

Où le nombre *0x68* représente l'adresse *I2C* du module *RTC*.

Ensuite, la date et l'heure dans le *RTC* sont mises à jour avec la ligne de code suivante :

```
ds3231.write_now()
```

La fonction *write_now()* stocke dans le module *RTC* la date et l'heure actuelles du système d'exploitation *Raspbian*.

Il existe une autre option pour stocker les données d'heure et de date. Cela peut être fait avec la ligne de code suivante :

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

Az-Delivery

La fonction appelée `write_all()` possède huit arguments et ne renvoie aucune valeur. Les arguments représentent une partie des données pour la date et l'heure. Les valeurs de tous les arguments sont des nombres entiers dans les gammes :

Seconds: 0 - 59, Minutes: 0 - 59, Hours: 0 - 23 Day: 1 - 7 (day in week),

Date: 1 - 31, Month: 1 - 12, Year: 0 - 99

Save_as_24h: True/False (tested with only as True)

Ensuite, une nouvelle fonction est créée, appelée `check()`. Cette fonction a un argument et renvoie une chaîne de caractères. Elle est utilisée pour ajouter le zéro initial au nombre à un chiffre. L'argument représente le nombre qui est ensuite vérifié s'il s'agit d'un nombre à un ou deux chiffres. Si le nombre est un chiffre simple, alors la valeur de chaîne `'0{}'.format(num)` est retournée. Si le nombre n'est pas un nombre à deux chiffres, la valeur de chaîne `str(num)` est renvoyée.

Après cela, le bloc de code `try-except` est créé. Dans le bloc de code `try`, la boucle indéfinie est créée. Dans cette boucle indéfinie, les données `RTC` sont lues et affichées dans le terminal. Il y a deux façons d'afficher les données dans le terminal. La première est d'utiliser la fonction `read_datetime()` et la seconde est d'utiliser la fonction `read_str()`.

Az-Delivery

Pour arrêter le script, appuyez sur '*CTRL + C*' sur le clavier. C'est ce qu'on appelle l'interruption du clavier. Lorsque l'interruption clavier se produit, le bloc de code précédent est exécuté, affichant le message *Script end !* dans le terminal.

Az-Delivery

La fonction `read_datetime()` a deux arguments et retourne un tuple. Le premier argument représente le siècle, une valeur entière, par exemple : pour le 21ème siècle, utilisez `century=21`. Le second argument représente l'information sur le fuseau horaire, utilisé pour les objets `datetime` (il n'est pas couvert dans cet eBook). La valeur de retour est un tuple qui comporte sept éléments. Les éléments sont : `dayOfWeek`, `dayOfMonth`, `month`, `year`, `hour`, `minute` et `second` (dans cet ordre). Toutes les valeurs de ces éléments sont des nombres entiers, sauf la valeur de l'argument `dayOfWeek` qui est une chaîne de caractères représentant le nom du jour de la semaine.

La fonction `read_str()` a un argument et renvoie une valeur de chaîne de caractères. L'argument est l'argument `century`, et est utilisé comme l'argument `century` de la fonction `read_datetime()`. La valeur de retour est une chaîne de caractères, avec un format prédéfini pour les données de date et d'heure, comme le montre le script suivant :

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

Temperature: 28.8°C

Pour obtenir des données de température à partir du

Az-Delivery

module *RTC*, utilisez la fonction appelée *detTemp()*. Cette fonction n'a pas d'arguments et renvoie une valeur flottante. La valeur de retour, une valeur flottante, représente les données de température en *Celsius*.

Il est maintenant temps d'apprendre et de réaliser vos propres projets. Vous pouvez le faire à l'aide de nombreux exemples de scripts et autres tutoriels, que vous trouverez sur Internet.

Si vous recherchez des microélectronique et accessoires de haute qualité, AZ-Delivery Vertriebs GmbH est l'entreprise idéale pour vous les procurer. Vous recevrez de nombreux exemples d'application, des guides d'installation complets, des livres électroniques, des bibliothèques et l'assistance de nos experts techniques.

<https://az-delivery.de>

Amusez-vous !

Mentions légales

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