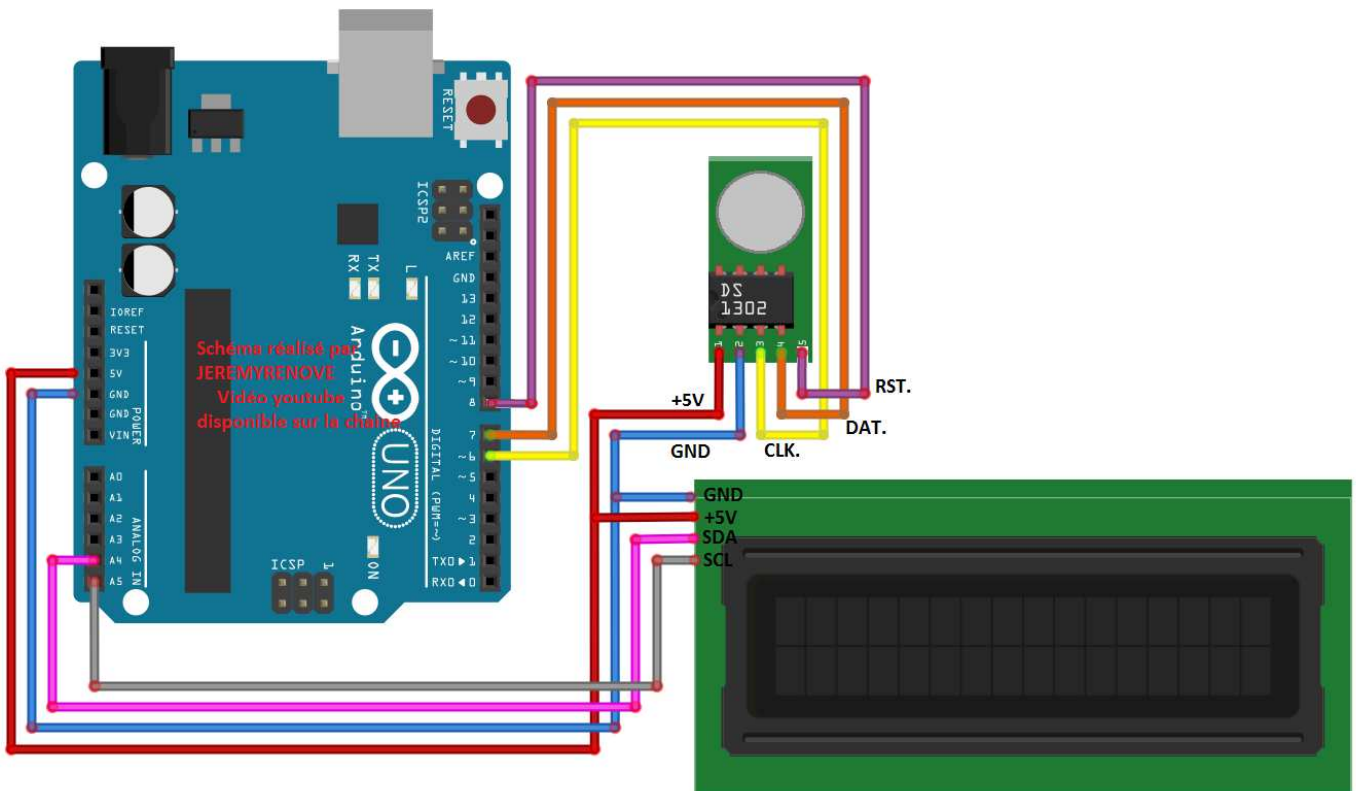


# Module RTC & Arduino

Le module RTC utilisé dans cet exemple sera de type DS1302, cependant, il existe d'autres modules similaires tels que le DS 1307. Dans le montage présenté, nous allons utiliser ce module pour construire une horloge. Montage le plus basique, cependant, dans notre série sur la domotique à 2€, il sera intéressant d'utiliser ce module pour réaliser un compteur d'énergie. En effet, pour calculer l'énergie, il faut multiplier une puissance pendant un temps.

Ce module Real time clock envoie une impulsion avec une grande précision et permet de garder en mémoire l'heure réglée même si une coupure de l'alimentation électrique survient.

Ci-dessous, voici le schéma de raccordement du module RTC (ici un DS 1302) :



Le montage se compose du module RTC, d'un arduino UNO et d'un écran LCD avec module I2C. Le +5Volts de l'arduino est raccordé à la borne +5V de l'écran et du module RTC, comme le montre le fil rouge. Le fil Bleu représente le GND qui se raccorde de l'arduino à l'écran et au module RTC. Ensuite, le module RTC comporte 3 autres points de raccordements, CLK (clock), Dat (la date) et Rst (reset). Le fil jaune fera la liaison entre la borne CLK et le pin digital 6. Le fil orange fera la liaison entre le pin DAT et le pin digital 7, et, le pin RST sera raccordé au pin digital 8 de l'arduino. Pour l'écran LCD et son module I2C, SDA se raccorde sur le pin A4, et SCL se raccorde sur le pin A5 de l'arduino.

Concernant le code, voici ce qu'il faut écrire dans l'IDE arduino pour faire fonctionner l'ensemble et afficher la date et l'heure sur l'écran LCD :

```
Fichier Édition Croquis Outils Aide
horloge_RTC
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <virtuabotixRTC.h> //Librarie a ajouter
virtuabotixRTC myRTC(6, 7, 8); //on declare les pin de cablage 6 pour l'heure, 7 pour la date, et 8 pour le reset

void setup() {
  Serial.begin(9600);
  lcd.init ();
  lcd.backlight();
  //pour la synchronisation, en premier les secondes, puis les minutes, heures, jour de la semaine, jour du mois, mois, puis année
  //myRTC.setDS1302Time(00, 46, 12, 6, 17, 11, 2018); //date et heure lors de la premiere synchronisation
  // la ligne ci dessus doit être mise en commentaire par la suite pour ne pas synchroniser l'heure a chaque televersement
}

void loop() {
  myRTC.updateTime();//mise a jour de la date et heure

  // on ecrit la date et l'heure dans le moniteur serie
  Serial.print(" Date / heure : ");
  Serial.print(myRTC.dayofmonth);
  Serial.print("/");
  Serial.print(myRTC.month);
  Serial.print("/");
  Serial.print(myRTC.year);
  Serial.print(" ");
  Serial.print(myRTC.hours);
  Serial.print(":");
  Serial.print(myRTC.minutes);
  Serial.print(":");
  Serial.println(myRTC.seconds);
  //on imprime les valeurs d'heure et temps sur l'ecran LCD
  lcd.setCursor(0, 0);
  lcd.print("heure:");
  lcd.setCursor(8, 0);
  lcd.print(myRTC.hours);
  lcd.setCursor(10, 0);
  lcd.print(" ");
  lcd.setCursor(11, 0);
  lcd.print(myRTC.minutes);
  lcd.setCursor(13, 0);
  lcd.print(" ");
  lcd.setCursor(14, 0);
  lcd.print(myRTC.seconds);|
  lcd.setCursor(0, 1);
  lcd.print("date:");
  lcd.setCursor(6, 1);
  lcd.print(myRTC.dayofmonth);
  lcd.setCursor(8, 1);
  lcd.print("/ ");
  lcd.setCursor(9, 1);
  lcd.print(myRTC.month);
  lcd.setCursor(11, 1);
  lcd.print("/ ");
  lcd.setCursor(12, 1);
  lcd.print(myRTC.year);

  // temporisation pour ralentir la vitesse de rafraichissement des données
  delay(1000);
}
Arduino/Genuine Uno sur COM1
```

Une fois recopié, il suffit de le téléverser le code pour faire fonctionner le montage. Attention à la ligne notée en gris (commentaire) sous la fonction du rétroéclairage. Il faut faire une première synchronisation de la date et l'heure pour le module, puis ensuite passer cette ligne en commentaire dans le code pour ne pas devoir synchroniser la date et l'heure à chaque mise à jour du code. Pour plus d'information sur le sujet, vous pouvez aussi regarder la vidéo traitant de ce sujet sur la chaine youtube jeremyrenove . Bon courage pour vos projets 😊