



#### **A RECEPTION DU COLIS :**

**Vérifier la composition** du colis indiquée ci-dessous

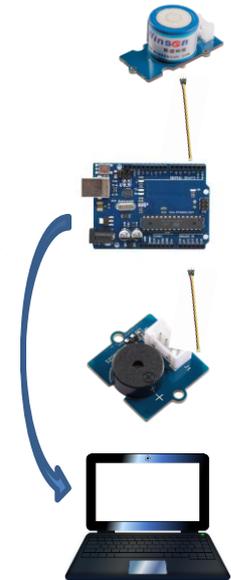
Kit complet pour démarrer en ARDUINO en SVT, comprenant:

- Les cartes UNO et SHIELD GROVE
- Un capteur température
- Un capteur humidité du sol
- Un capteur de lumière
- Un boîtier transparent pour carte UNO
- Boîtier de rangement compartimenté
- Câble d'alimentation USB
- Pile 9 V et coupleur de pile

Les capteurs sont livrés avec leur câble.

## INTRODUCTION AUX MICROCONTROLEURS

- Un microcontrôleur est un circuit intégré contenant les éléments essentiels d'un ordinateur (processeur, mémoire, unités périphériques et interfaces d'entrées et de sorties).
- Il est de petite taille, présente une faible consommation en énergie et un coût réduit.
- C'est ce qu'on appelle souvent 'Système embarqué' (dans les moteurs automobiles, les Smartphones, l'électroménager, les jouets...)
- Avec un système de microcontrôleur, la chaîne d'acquisition des données expérimentales devient ainsi complètement transparente pour les élèves :
  - Capteurs
  - Microcontrôleurs (lecture de la donnée)
  - Pilotage d'un élément (action d'un ventilateur ou d'une alarme...)
  - Analyse ou enregistrement des données (ordinateur)

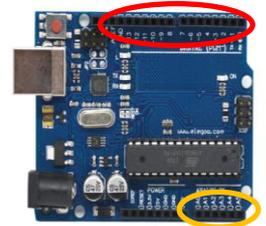


### La carte Type Arduino UNO

Elle dispose de :

- 14 entrées/sorties numériques
- 6 entrées analogiques
- Un oscillateur à quartz de 16 MHz
- Une connexion USB
- Une prise d'alimentation
- Un bouton de reset

Elle contient tout le nécessaire pour piloter le microcontrôleur, il suffit de la connecter à un ordinateur avec un câble USB (fourni) pour l'utiliser simplement. Elle peut donc être utilisée telle quelle, mais les branchements peuvent paraître délicats avec beaucoup de petits câbles à manipuler.



### La platine Shield Grove

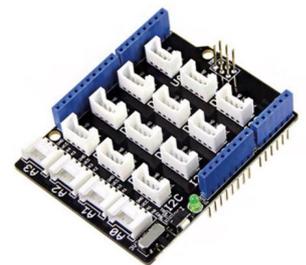
Afin de faciliter l'usage de la carte Arduino, il est conseillé d'ajouter une platine GROVE qui se positionne sur toute carte type Arduino Uno.

Elle simplifie grandement les branchements des capteurs :

Câbles équipés de connecteurs rapides et détrompeurs : impossible de se tromper et moins de câbles 'en vrac'.

Les branchements tiennent mieux.

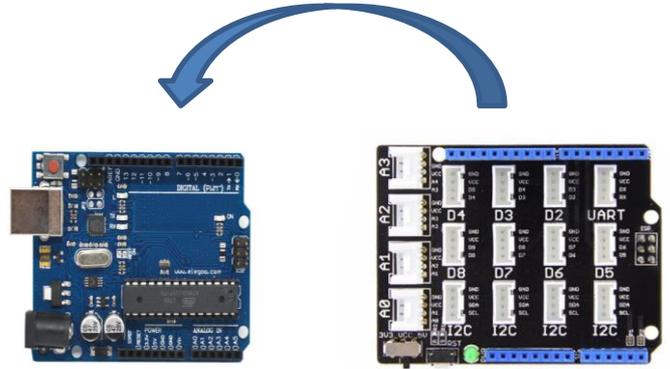
On utilise les entrées/sorties de la carte UNO mais en se connectant sur la carte Grove.



## MISE EN SERVICE DU SYSTEME

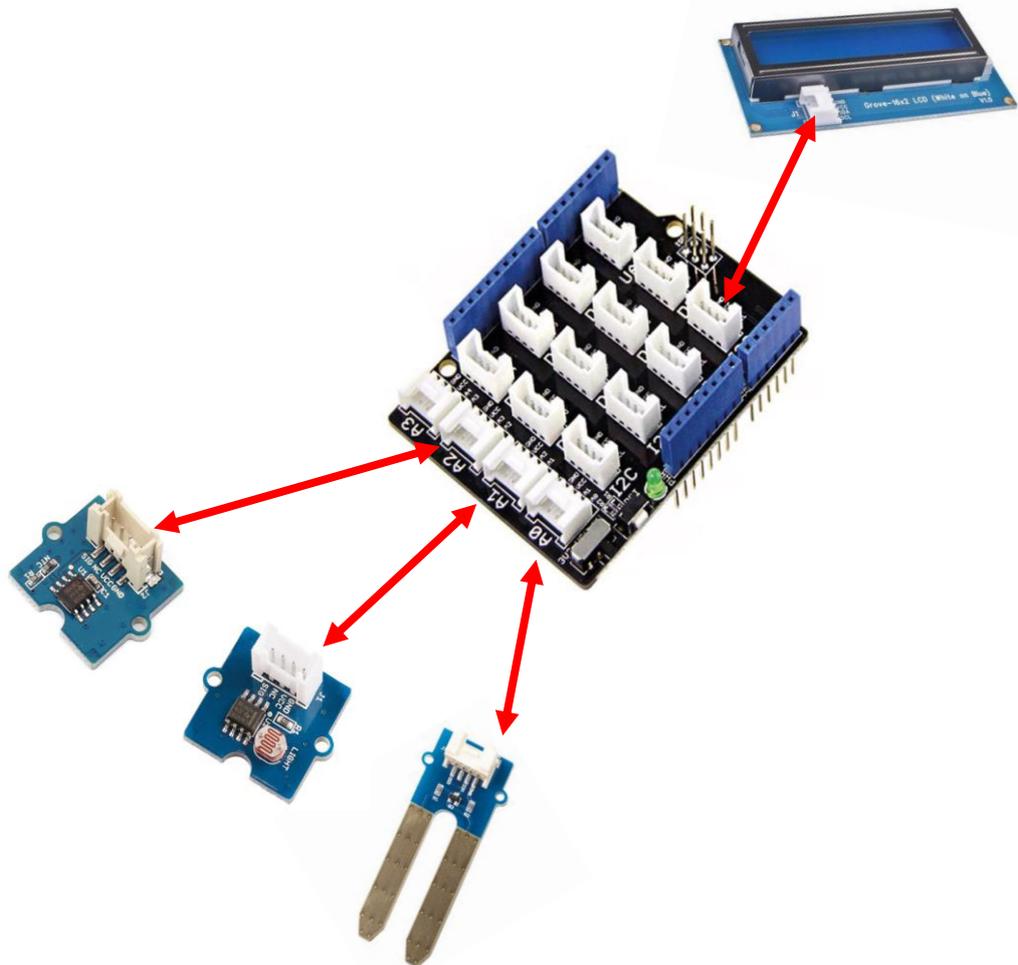
Assembler les cartes :

Placer la carte Grove Shield sur la carte type UNO



## PROPOSITION DE BRANCHEMENTS DES CAPTEURS

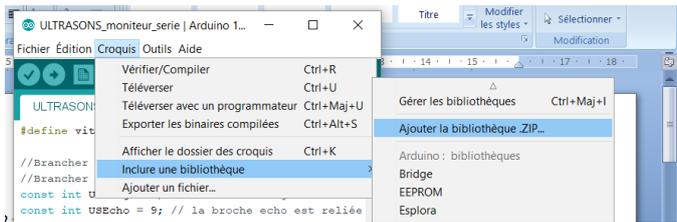
1. Brancher le capteur lumière sur l'entrée analogique A2
2. Brancher le capteur température sur l'entrée analogique A1.
3. Brancher le capteur d'humidité du sol sur l'entrée analogique A0.
4. Brancher l'écran sur l'entrée I2C.
5. Brancher l'alimentation générale de la carte sur l'ordinateur grâce au câble livré ou sur la pile . Le boîtier de contrôle de la serre peut également être connecté à un ordinateur ou alimenté avec une pile 9V.
6. Dès que le boîtier (et donc le microcontrôleur) sera sous tension, l'écran.



## INSTALLATION DU LOGICIEL ET PROGRAMMATION

Installer le logiciel ARDUINO, pour cela, télécharger le logiciel en allant sur la page article sur notre site : <https://sordalab.com/FR/catalogue/svt/kit-nouveaux-programmes-type-arduino-uno-svt,ARDUSVT,.php> ou en tapant ARDUSVT dans le moteur de recherche sur notre site SORDALAB.COM.

- Décompresser le dossier du logiciel
- Installer le logiciel en double cliquant sur le fichier ARDUINO.EXE.
- Télécharger également le dossier 'Librairie écran LCD'
- Lancer le logiciel
- Inclure la bibliothèque en allant dans 'Croquis/Inclure une bibliothèque.ZIP'



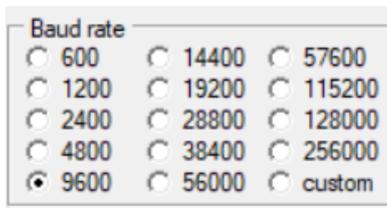
- Vous pouvez à présent travailler sur le programme contenu dans cette notice en le copiant dans le logiciel. Ainsi vous pouvez modifier ce que vous souhaitez. Puis 'téléversez' le dans le microcontrôleur ' (qui doit être connecté à l'ordinateur).

### Pour aller plus loin :

- Basculer les informations de la carte vers Excel en temps réel :
- Télécharger le logiciel Terminal.exe disponible sur la page de ARDUSVT sur SORDALAB.COM
  - Lancer l'appli Terminal.exe du dossier « A INSTALLER OU INSERER »
  - Ouvrir un fichier « sketch » utilisant le moniteur série
  - Remplacer les « : » du texte par « ; » pour faciliter l'enregistrement en fichier
  - Connecter la carte Arduino et retenir le port COM utilisé
  - Sélectionner le bon port COM et si le port n'apparaît pas dans la liste déroulante, cliquer sur rescan



- Vérifier la fréquence rentrée dans le code (le plus souvent 9600), modifier si nécessaire



- Retourner sur le logiciel Arduino et téléverser
- Cliquer sur 
- Cliquer sur  pour lancer l'enregistrement.

- Si nécessaire appuyer sur le reset de la carte et sur Clear pour répartir à zéro
- Cliquer  sur pour arrêter l'enregistrement
- Choisir, « tout type de fichier » dans la boîte de dialogue qui s'ouvre pour l'enregistrement. Choisir l'emplacement souhaité, nommé le fichier enregistré et ajouter l'extension « .csv »
- Ouvrir le fichier sous Excel puis modifier les. En, pour pouvoir créer des graphes ou des histogrammes

### **EXEMPLE DE PROGRAMME incluant capteur d'humidité, lumière et température**

Les `'/'` en début de lignes rendent les lignes inopérantes, ce qui permet de mettre un commentaire, ou d'inactiver une partie du programme. Il suffit de les supprimer pour réactiver les lignes concernées.

```
#include <rgb_lcd.h>
#include <math.h>
```

#### **//Les capteurs et accessoires**

```
int moistureSensor = A0; // La broche d'entrée pour le capteur d'humidité
int lightSensor = A2; // La broche d'entrée pour le capteur de lumière
```

#### **//Paramétrage de la valeur de température**

```
const int B = 4275; // B value of the thermistor
const int R0 = 100000; // R0 = 100k
const int pinTempSensor = A1; // Le capteur de température est connecté à A1
```

```
#include <Wire.h>
#include "rgb_lcd.h"
rgb_lcd lcd;
const int colorR = 255;
const int colorG = 0;
const int colorB = 0;
```

#### **/\*\* Attention : Sélectionnez correctement le protocole de communication**

```
void setup()
```

```
{
  Serial.begin(9600); // Afficher la sortie sur le port série
  lcd.begin(16, 2); // définir le nombre de colonnes et de lignes de l'écran LCD
  lcd.setRGB(colorR, colorG, colorB); // L'une des fonctions les plus importantes de Grove - LCD RGB
  Backlight est le changement de la couleur du rétroéclairage.
```

```
  delay(1000);
}
```

**void loop()**

{

**// mettez votre code principal ici, pour l'exécuter de façon répétée :**

int HUMIDITE = analogRead(moistureSensor); // lire la valeur de la sonde d'humidité :

int LUMIERE = analogRead(lightSensor); // lire la valeur de la sonde de lumière

int a = analogRead(pinTempSensor); // lire la valeur de la sonde température

float R = 1023.0 / a - 1.0;

R = R0 \* R;

float temperature = 1.0 / (log(R / R0) / B + 1 / 298.15) - 273.15;

// convertit la valeur lue en température (voir fiche technique du capteur)

int TEMP = (temperature);

**//Afficher les valeurs sur l'écran LCD**

lcd.print("Capteurs"); // Affiche un titre, ici 'Serre pilotée' (NB : ne pas mettre d'accents)

lcd.setCursor(0, 1); // placer le curseur sur la colonne 0, ligne 1 (la ligne1 est la deuxième ligne)

lcd.print("Temp "); // Affiche 'Temp' avant la valeur mesurée

lcd.print(TEMP); //Affichage de la valeur de température

lcd.print(" Hum "); // Affiche 'Hum' avant la valeur mesurée

lcd.print(HUMIDITE); //Affichage de la valeur d'humidité du sol

lcd.print(" Lux"); // Affiche 'Lux' avant la valeur mesurée

lcd.print(LUMIERE); //Affichage de la valeur de la lumière

**//Afficher les valeurs sur le moniteur série (ordinateur)**

Serial.print(" Temp "); // Affiche 'Temp' avant la valeur mesurée

Serial.println(TEMP); //Affichage de la valeur de température

Serial.print(" Hum "); // Affiche 'Hum' avant la valeur mesurée

Serial.println(HUMIDITE); //Affichage de la valeur d'humidité du sol

Serial.print(" Lux "); // Affiche 'Lux' avant la valeur mesurée

Serial.println(LUMIERE); //Affichage de la valeur de la lumière

delay(2000);

}

**CODES FOURNIS PAR LE FABRICANT POUR CHAQUE CAPTEUR:**

- **Capteur température**

#include <math.h>

const int B = 4275; // B value of the thermistor

const int R0 = 100000; // R0 = 100k

const int pinTempSensor = A0; // Grove - Temperature Sensor connect to A0

#if defined(ARDUINO\_ARCH\_AVR)

#define debug Serial

#elif defined(ARDUINO\_ARCH\_SAMD) || defined(ARDUINO\_ARCH\_SAM)

```
#define debug SerialUSB

#else
#define debug Serial
#endif

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int a = analogRead(pinTempSensor);
  float R = 1023.0/a-1.0;
  R = R0*R;
  float temperature = 1.0/(log(R/R0)/B+1/298.15)-273.15; // convert to temperature via datasheet
  Serial.print("temperature = ");
  Serial.println(temperature);
  delay(100);
}
```

- **Capteur lumière**

```
#include <Grove_LED_Bar.h>
Grove_LED_Bar bar(3, 2, 0); // Clock pin, Data pin, Orientation
```

```
void setup()
{
  // nothing to initialize
  bar.begin();
  bar.setGreenToRed(true);
}
```

```
void loop()
{
  int value = analogRead(A0);
  value = map(value, 0, 800, 0, 10);
  bar.setLevel(value);
  delay(100);
}
```

- **Capteur humidité**

```
int sensorPin = A0;
int sensorValue = 0;
```

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.print("Moisture = " );
  Serial.println(sensorValue);
}
```

```
delay(1000);  
}
```

- **L'écran**

```
#include <Wire.h>  
#include "rgb_lcd.h"  
rgb_lcd lcd;  
/*  
const int colorR = 255;  
const int colorG = 0;  
const int colorB = 0;  
*/  
  
void setup()  
{  
// set up the LCD's number of columns and rows:  

```