Mayar , Ruby-Joe : Prototypage

Contexte

Dans le cadre de nos expérimentations avec l'Arduino, nous avons exploré son potentiel en développant un système simple de détection d'humidité.

Objectif

Utiliser un capteur d'humidité avec une carte Arduino pour contrôler des LEDs. Lorsque l'humidité dépasse un **seuil de 50%**, la **LED rouge** s'allume. Si l'humidité est inférieure à ce seuil, c'est la **LED verte** qui s'allume.

Matériel

- 1 carte Arduino
- 1 BASE SHIELD
- 2 résistances
- 1 plaque de montage rapide (breadboard)
- 1 LED verte
- 1 LED rouge
- 4 fils de connection
- 1 capteur de température et d'humidité (type SHT31 ou similaire)

Montage Physique

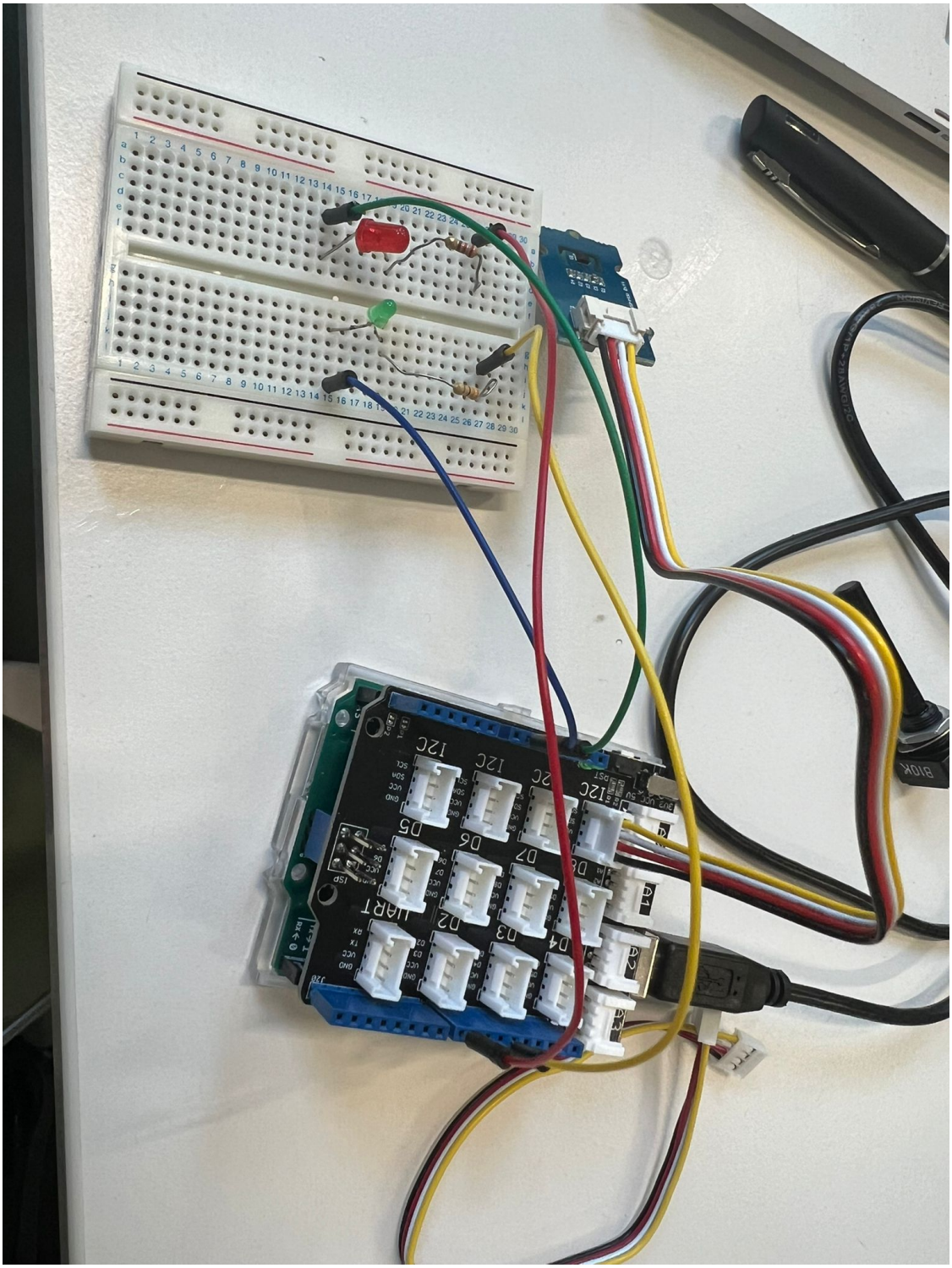
Le capteur de température et d'humidité est connecté à la carte **Arduino** via un **Base Shield**, facilitant l'interface avec le capteur en utilisant le **protocole I2C**. Le rôle de ce capteur est de mesurer l'humidité ambiante.

Deux LEDs (une rouge et une verte) sont utilisées pour indiquer le niveau d'humidité:

- La **LED rouge** s'allume lorsque l'humidité **dépasse 50%**.
- La **LED verte** s'allume lorsque l'humidité est **inférieure à 50%**.

Les deux LEDs sont connectées à la **breadboard**, chacune avec une résistance pour **limiter** le courant. La **LED rouge** est branchée sur la **broche 8** de l'Arduino, et la **LED verte** sur la **broche 12**. Elles sont ensuite reliées à la **masse (GND)** de l'Arduino pour fermer le circuit.

Le tout est alimenté par **l'Arduino**, qui est relié à un **ordinateur** via **USB** pour à la fois alimenter le circuit et afficher les résultats sur le moniteur série.



Explication du Code

Le code se compose de plusieurs parties importantes:

1. **Inclusion des bibliothèques et initialisation des variables**
   Ici, on inclut les bibliothèques nécessaires pour utiliser le protocole I2C et le capteur d'humidité SHT31. Les pins des LEDs sont définies respectivement sur les **broches 8**

(LED rouge) et **12 (LED verte)**. Les variables **start** et **stop** sont initialisés et utilisées pour mesurer le temps d'exécution de la lecture du capteur en microsecondes. L'objet **sht** permet d'interagir avec le capteur SHT31.

```
#include "Wire.h"
#include "SHT31.h"
#define LEDROUGE 8
#define LEDVERTE 12
#define SHT31_ADDRESS   0x44

uint32_t start;
uint32_t stop;

SHT31 sht;
```

2. **Configuration initiale dans la fonction `setup()`**
   Cette partie du code initialise la communication série à 9600 baud ***(Serial.begin(9600))*** pour afficher les résultats sur le moniteur série. Les pins des LEDs sont configurées en tant que sorties, et la communication I2C ***(Wire.begin())*** avec le capteur est démarrée.

```
void setup()
{
  Serial.begin(9600);
  pinMode(LEDROUGE, OUTPUT);
  pinMode(LEDVERTE, OUTPUT);
  Serial.println(__FILE__);
  Serial.print("SHT31_LIB_VERSION: \t");
  Serial.println(SHT31_LIB_VERSION);

  Wire.begin();
  Wire.setClock(100000);
  sht.begin();

  uint16_t stat = sht.readStatus();
  Serial.print(stat, HEX);
  Serial.println();
}
```

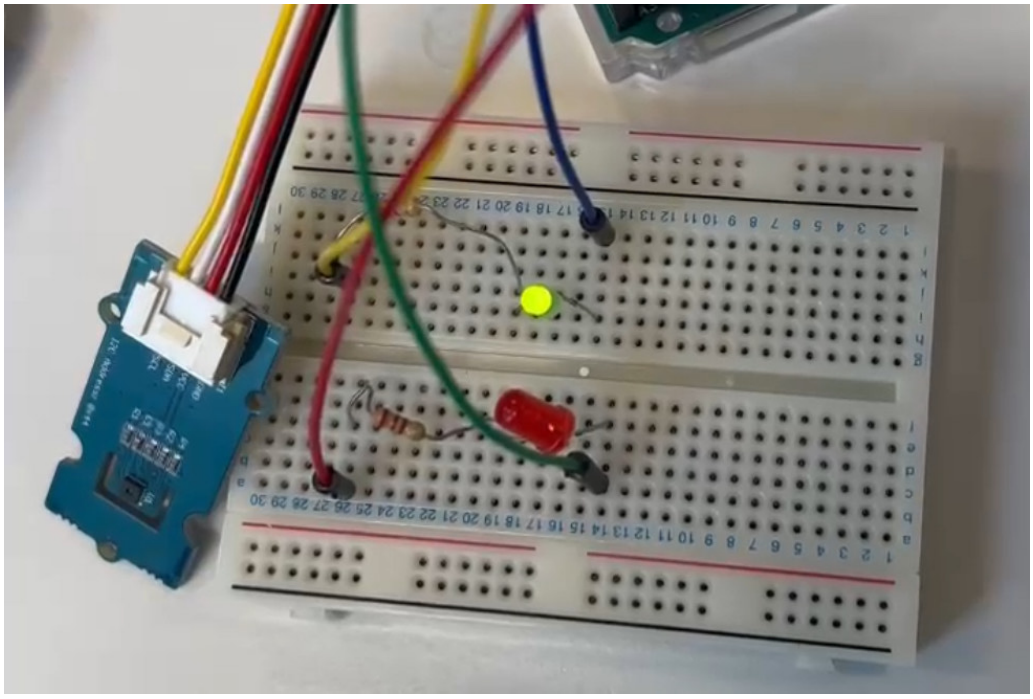
3. **Lecture des données d'humidité et Condition de contrôle des LEDs**
   Dans la boucle principale, le capteur d'humidité est interrogé, et la valeur mesurée est récupérée et affichée sur le moniteur série. Le capteur renvoie un pourcentage d'humidité.
   Ensuite, avec une condition **"*if*"**, selon le niveau d'humidité mesuré, le programme allume la **LED rouge** ou la **LED verte.** Si l'humidité **dépasse** ou est **égale à 50%**, la **LED rouge** s'allume (**digitalWrite(LED, HIGH)**) et la **LED verte** s'éteint. Si l'humidité est **inférieure à 50%**, c'est la **LED verte** qui s'allume et la **LED rouge** qui s'éteint. Ce contrôle est répété à chaque itération de la boucle.
   La fonction **delay(100)** introduit une pause de 100 millisecondes entre chaque lecture pour éviter une boucle trop rapide.

```
void loop()
{
  start = micros();
  sht.read();         //  default = true/fast       slow = false
  stop = micros();
  float humidity=sht.getHumidity();

  Serial.print("Humidity:");
  Serial.println(humidity);
  if (humidity>=50){
    digitalWrite(LEDROUGE, HIGH);  // turn the RED LED on (HIGH is the voltage level)
    digitalWrite(LEDVERTE, LOW);  // turn the GREEN LED off (LOW is the voltage level)

  }
  else{
    digitalWrite(LEDVERTE, HIGH);  // turn the GREEN LED on (HIGH is the voltage level)
    digitalWrite(LEDROUGE, LOW);   // turn the RED LED off by making the voltage LOW
  }

  delay(100);     // wait for a 0.1 second
}
```

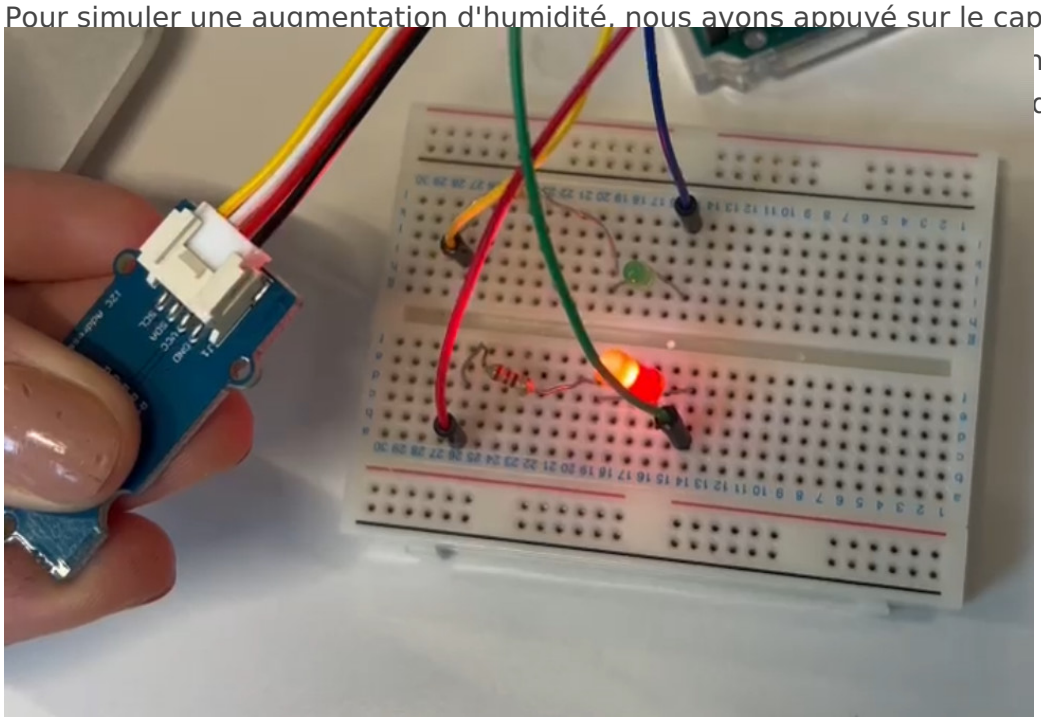
Tests

Après avoir terminé le montage et la programmation de l'Arduino, nous avons effectué des tests pour vérifier le bon fonctionnement du système.

1. **Test 1 : Humidité inférieure à 50%**
   Lorsque le capteur détecte une humidité **inférieure à 50%**, la **LED verte** s'allume, indiquant que le niveau d'humidité est dans la plage normale qu'on a indiqué.



2. **Test 2 : Humidité supérieure à 50%**
   Pour simuler une augmentation d'humidité, nous avons appuyé sur le capteur avec notre
   humidité
   dité est élevée.



Revision #14
Created 21 September 2024 18:34:16 by Abdelgawad Mayar
Updated 23 September 2024 00:50:55 by Abdelgawad Mayar