

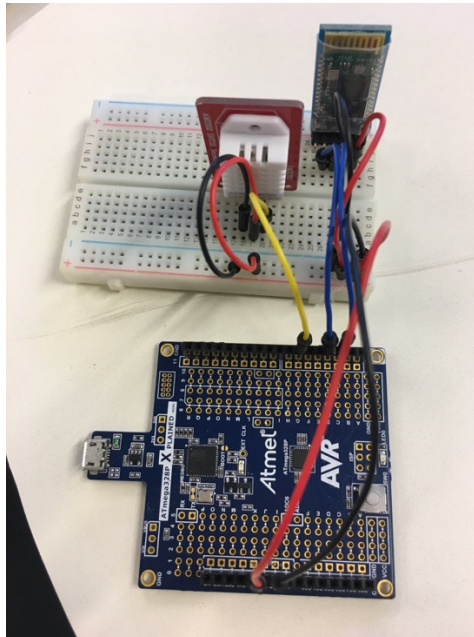
Rapport

Pendant la semaine de pause, après avoir fait le montage j'ai essayé de faire fonctionner le module Bluetooth HM-10 adapté aux iPhone mais mon programme n'a pas fonctionné.

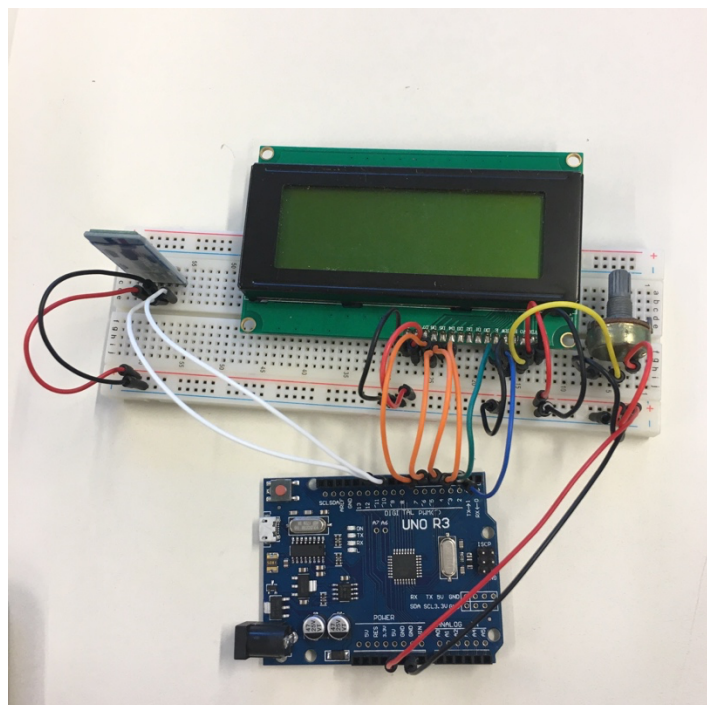
On a alors décidé avec mon binôme que je devais arrêter avec le module cité précédemment car il fallait de toute façon continuer à avancer dans notre projet et ne pas rester bloquer.

Du coup pendant la séance de TD j'ai repris les anciens modules Bluetooth, le HC05 (esclave) et le HC06 (maître), et j'ai fait les différents montages : un pour la télécommande et un pour le rover. Le montage pour la télécommande du rover est composé d'un écran LCD 20x4 et du module Bluetooth HC05, et le montage qui sera dans la voiture est composé du capteur de température DH22 et du module Bluetooth HC06.

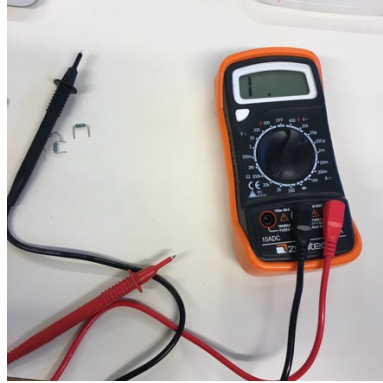
Montage qui sera posé sur le rover :



Montage qui sera posé sur la télécommande :



De plus, je pensais au début que je devais forcément mettre une résistance pour le Bluetooth mais comme je ne connaissais pas les résistances qu'on possédait (combien d'ohm), j'ai dû utiliser un voltmètre pour la première fois et j'ai trouvé ça plutôt pratique (encore une nouvelle petite découverte).



Ensuite après mes deux montages, j'ai commencé à écrire les différents programmes en arduino de mes composantes.

Le programme pour le capteur de température DH22 (j'ai dû enregistrer une nouvelle librairie pour que ça puisse fonctionner – *DHT sensor library*) :

```
#include <DHT.h>
```

```
#define DHTPIN 6 // branchée sur la pin D6
```

```
// Le type de DHT utilisé
```

```
#define DHTTYPE DHT22 // ici on utilise DHT22
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
//Initialisation du programme
```

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("Pour tester");  
  Serial.println();  
  dht.begin();  
}
```

```
//Boucle principale
```

```
void loop() {  
  // Lecture des données  
  float tauxHumidite = dht.readHumidity(); // Lecture du taux d'humidité zn pourcentage  
  float tempCelsius = dht.readTemperature(); // Lecture de la température en degrés Celsius
```

```
// Vérification
```

```
if (isnan(tauxHumidite) || isnan(tempCelsius)) {  
  Serial.println("Aucune valeur retournée. vérifier le branchement");  
  delay(2000);  
  return; // Si aucune valeur reçu, on a une attente de deux secondes, puis redémarrage de cette fonction  
}
```

```
// Calcul de la température ressentie
```

```
float tempResentieCelsius = dht.computeHeatIndex(tempCelsius, tauxHumidite, false); // le false permet de montrer qu'on est en celcius et non en fahrenheit
```

```
// Affichage des valeurs
```

```
Serial.print("Humidité = "); Serial.print(tauxHumidite); Serial.println(" %");  
Serial.print("Température = "); Serial.print(tempCelsius); Serial.println(" °C");  
Serial.print("Température ressentie = "); Serial.print(tempResentieCelsius); Serial.println(" °C");  
Serial.println();  
delay(2000);  
}
```

Celui-ci a fonctionné à la fin de la séance, et j'ai commencé à écrire le programme d'un des module Bluetooth.

Je vais continuer décrire les autres programmes d'ici la fin de la semaine et les combiner pour voir si tout fonctionne correctement.