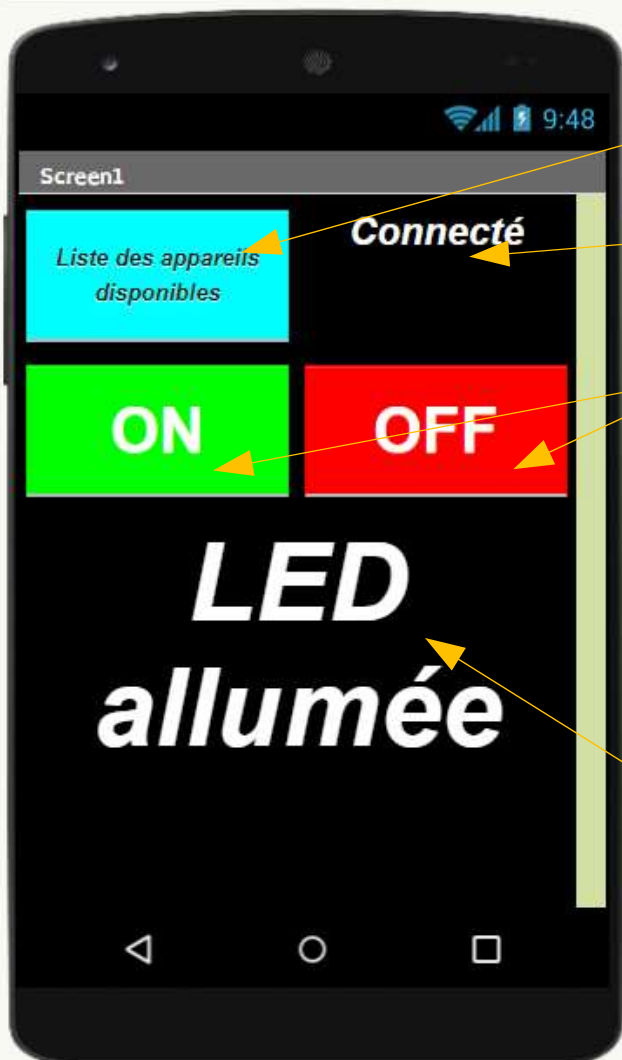


## COMPTE RENDU :

### Liste des composants nécessaires pour réaliser votre système

- récepteur Bluetooth
- module relais
- carte Arduino
- fils électriques
- bread born
- ampoule
- générateur (alimentation)
- câbles spéciaux

### -Application de commande du système d'éclairage (MIT App Inventor)



Ce bouton permet de choisir l'appareil qui sera connecté en Bluetooth.

Indication sur l'état connecté ou déconnecté du téléphone sur le Bluetooth.

Ces deux boutons permettent d'allumer (On) et éteindre (OFF)

Indication sur l'état de la lampe

Composants non-visible



BluetoothClient1

Programme utilisée pour l'application :

Programme pour la fonction Bluetooth :

```
quand ListPicker1 .Avant prise
faire mettre ListPicker1 . Eléments à BluetoothClient1 . Adresses et noms

quand ListPicker1 .Après prise
faire si appeler BluetoothClient1 .Se connecter
        adresse ListPicker1 . Sélection
alors mettre Label1 . Texte à "Connecté"
sinon mettre Label1 . Texte à "Deconnecté"
```

Programme pour la fonction Led allumer et éteindre :

```
quand Button1 .Clic
faire mettre Label2 . Texte à "LED allumée"
    si BluetoothClient1 . Est connecté
alors appeler BluetoothClient1 .Envoyer texte
        texte " A "

quand Button2 .Clic
faire mettre Label2 . Texte à "LED éteinte"
    si BluetoothClient1 . Est connecté
alors appeler BluetoothClient1 .Envoyer texte
        texte " E "
```

## -Programmes d'exécution de la commande (Arduino) (avec les commentaires)

```
void setup()
{
  // Nous "annonçons" sur quel pin nous allons travailler (ici le 13):
  pinMode(13,OUTPUT);
  // Ici nous lançons la communication série par cette commande, en 9600 bit par seconde:
  Serial.begin(9600);
}

void loop() {
  /* La variable de notre programme est celle ci-dessous, donc soit elle est égale à
  'A', soit à 'E'selon l'information reçue:*/
  switch(Serial.read())
  {
    // Si le signal reçue est 'A', alors la LED branchée au pin 13 s'allume:
    case 'A' : digitalWrite(13,HIGH);
      break; // Arrêt de la recherche de conditions si variable=A

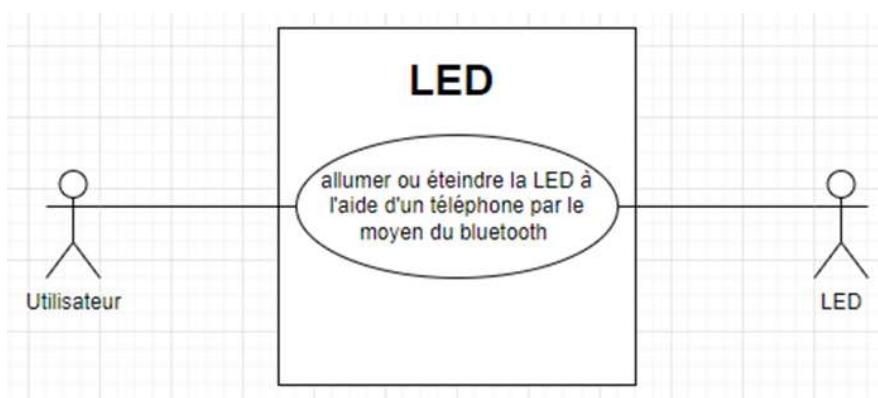
    // Si le signal reçue est 'E', alors la LED branchée au pin 13 s'éteint:
    case 'E' : digitalWrite(13,LOW);
      break; // Arrêt de la recherche de conditions si variable=E
  }
}
```

## Explications du fonctionnement du système avec du texte, croquis et des images légendées...

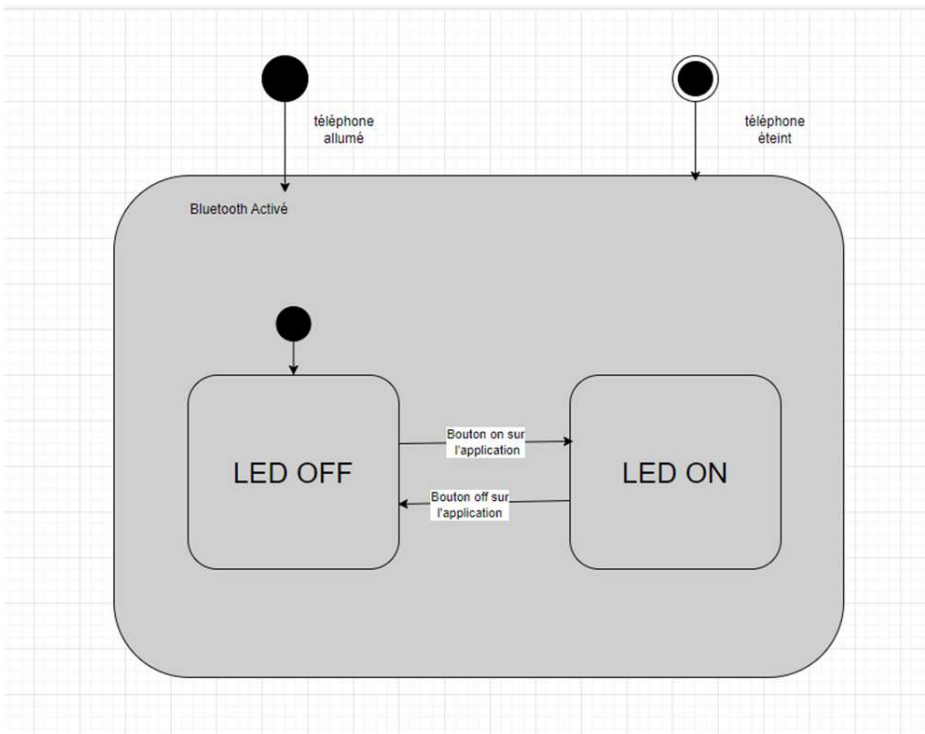
Nous avons commencé par faire deux groupes :

Un binôme pour l'application écrire le programme et faire le design de l'application.

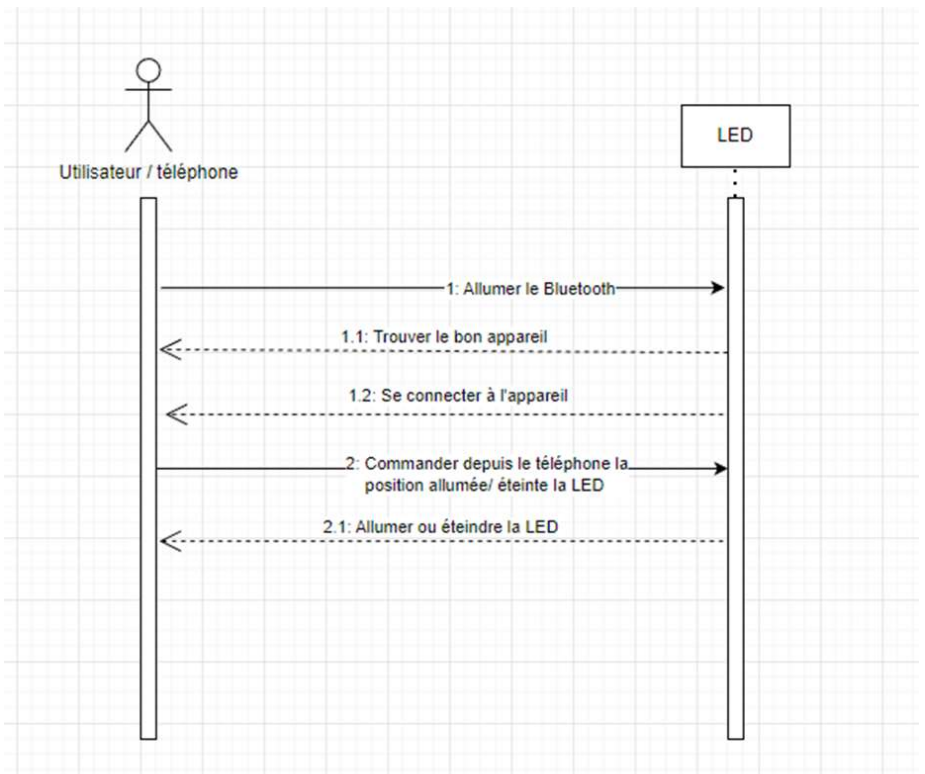
Le deuxième binôme pour faire les diagrammes :



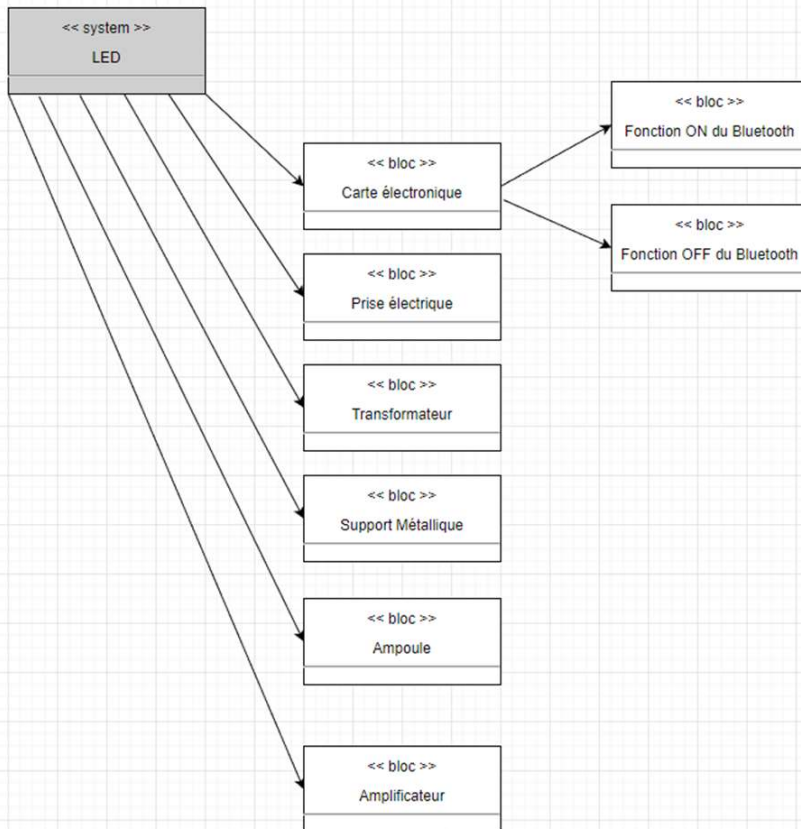
C'est le diagramme des cas d'utilisation qui permet d'avoir la fonction principale du système.



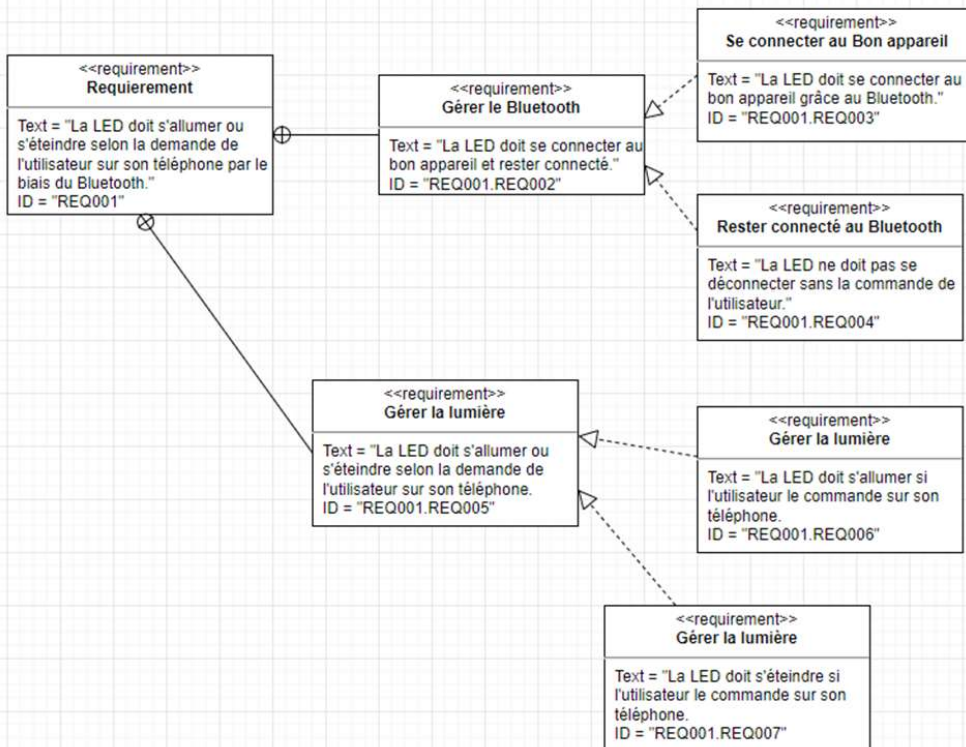
C'est le diagramme d'état qui permet de représenter des objets liés qui permettent d'obtenir le fait que la lampe soit allumée ou éteinte.



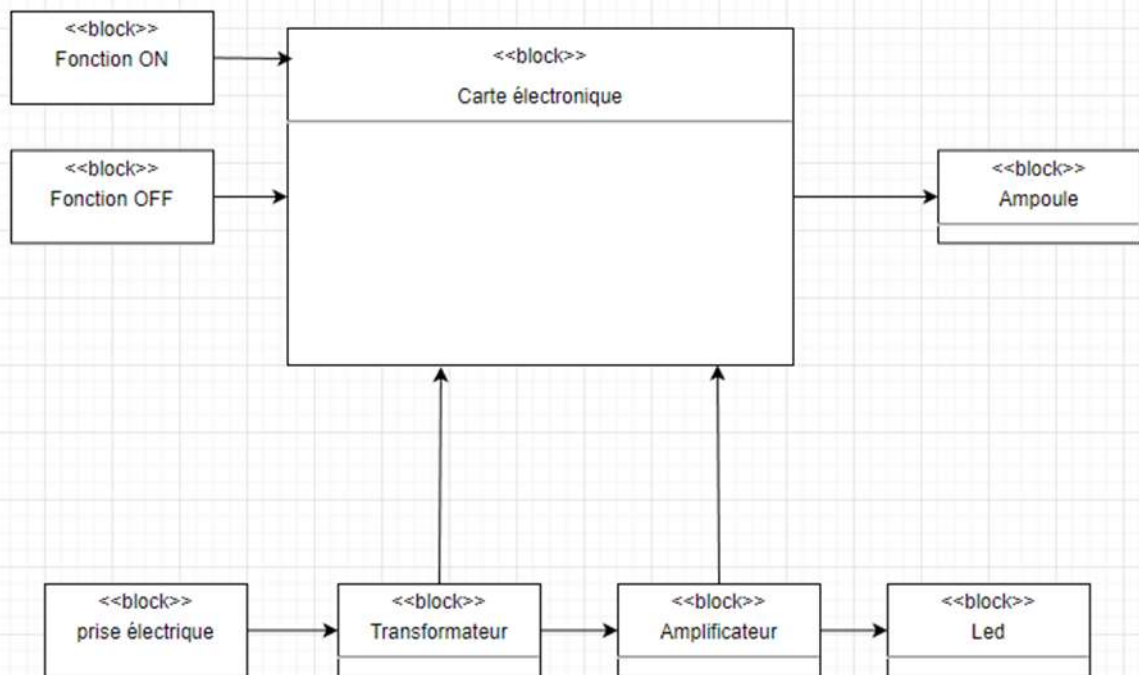
Le diagramme de séquence permet de voir les fonctionnalités du système.



Le diagramme de définitions des blocs nous a permis de savoir les composants, nous avons besoin pour répondre aux fonctionnalités souhaitées.



Le diagramme des exigences nous a permis de savoir et de classer les fonctionnalités dont nous avons besoin pour réussir le projet.



Le diagramme des blocs internes nous a permis de définir les échanges d'information, d'énergie et de matière entre les blocs du système pour son bon fonctionnement.

Ensuite, nous avons fait du Proteus pour modéliser et er du Arduino pour coder les informations que l'on voulait mettre dans la carte et enfin nous avons réalisé le montage qui a bien fonctionné.

Le fonctionnement du système est que l'humain appuie sur un bouton (on ou off) sur l'application qui envoie un message en Bluetooth qui est reçu par le récepteur Bluetooth qui lui ensuite transférer l'information à la carte Arduino qui la traite. La carte Arduino envoie un message au module relais qui lui en fonction du message laisse passer le courant, ce qui allume ou non la Led.

### Plan de travail :

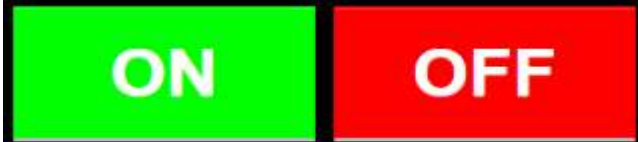
#### Premier jour (2H au compteur) :

Nous avons réfléchi sur le sujet du projet à réaliser à la fin des 12 heures et à comment le réaliser. Nous avons commencé à s'exercer sur MIT App Inventor afin de comprendre comment créer notre application. Et nous avons fait la réalisation d'un diagramme sur les 7 sur draw.io (diagramme des cas d'utilisation).

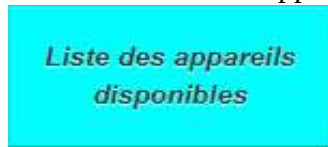
#### Deuxième jour (4H au compteur) :

On a terminé les 7 diagrammes et commencé l'application sur MIT App Inventor :

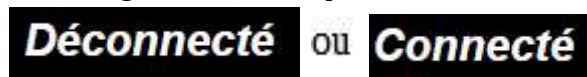
- 2 boutons ( ON et OFF ) afin d'allumer et éteindre la LED



- 1 bouton « liste des appareils disponibles » afin de détecter la LED



- 1 affichage ou il est marqué « Déconnecté » ou « Connecté » selon la situation.



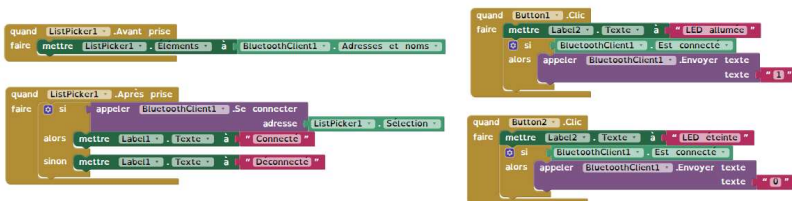
- 1 affichage pour savoir si la LED est éteinte ou allumée



#### Troisième jour (6H au compteur) :

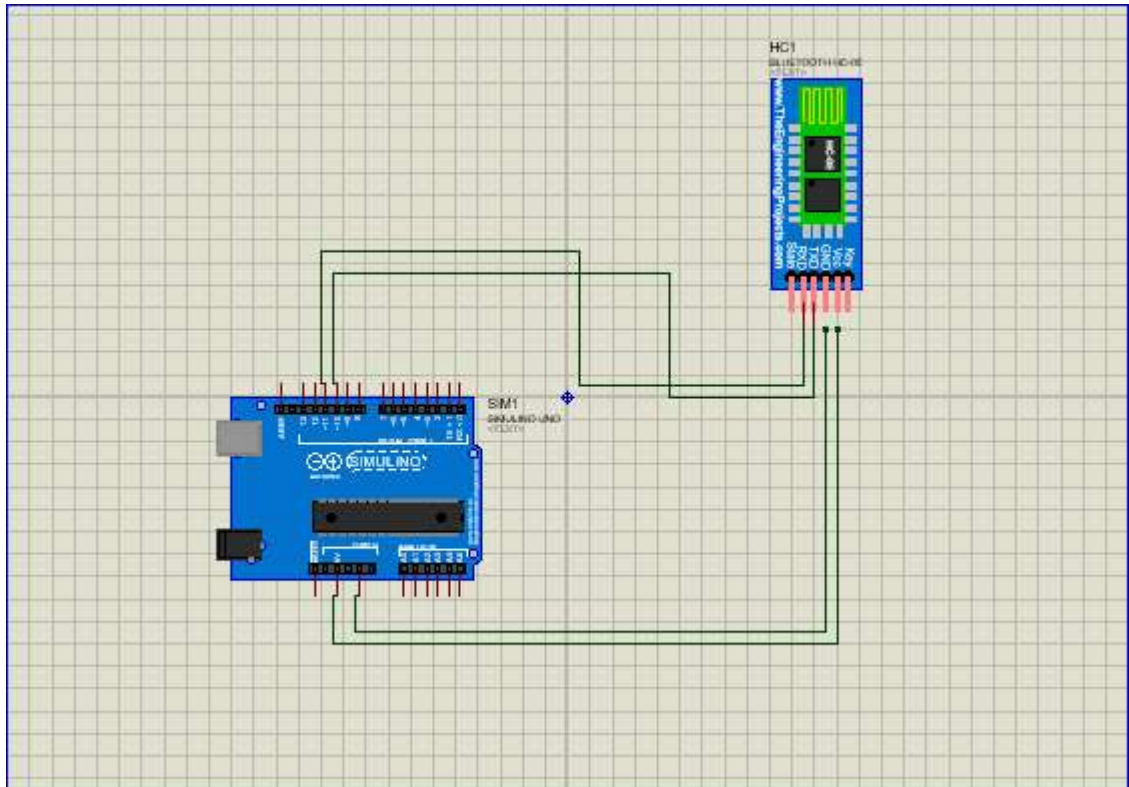
On a presque terminé l'application sur MIT App Inventor et on a commencé le programme sur Arduino. On a aussi modélisé notre montage sur Proteus.

- Voici notre code sur MIT App Inventor :



#### Quatrième jour (8H au compteur) :

On a terminé complètement l'application. On a du recommencer le codage Arduino et nous l'avons terminé. Nous avons modélisé sur Proteus.

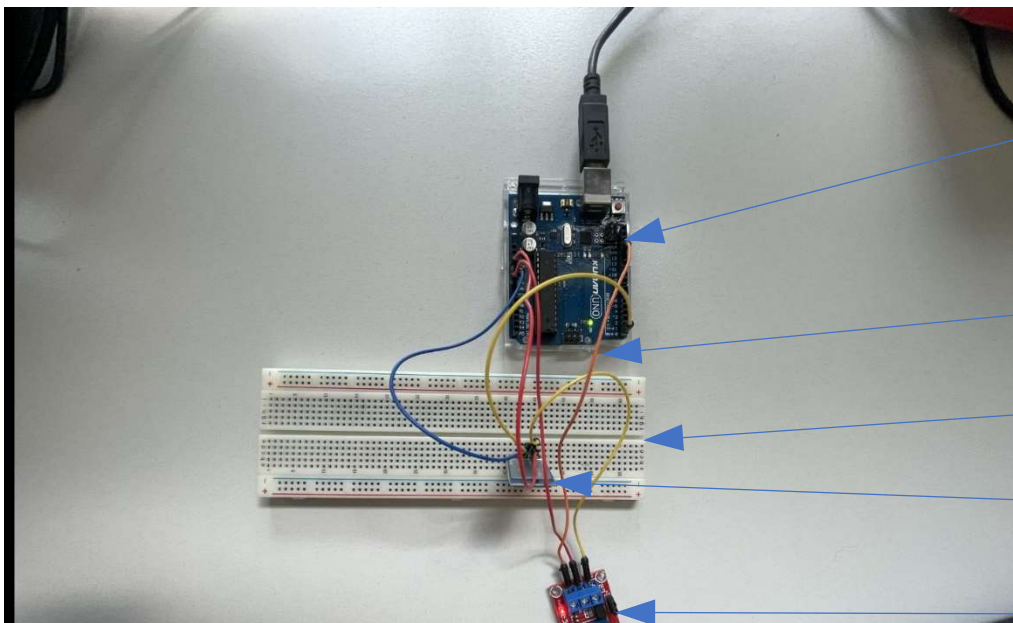


**Cinquième jour (10H au compteur) :**

Nous avons commencé à faire le circuit électrique. Nous avons eu du mal au début à trouver le bon circuit car nous avons eu du mal à trouver les informations. Nous avons commencé par brancher entre le récepteur Bluetooth et la carte Arduino. C'est la première que nous avons fait sur Proteus.

**Sixième jour (12H au compteur) :**

Nous avons fini de réaliser le circuit et nous avons fait la vidéo.



Carte Arduino

Fils électriques

Bread born

Récepteur Bluetooth

Module relais