

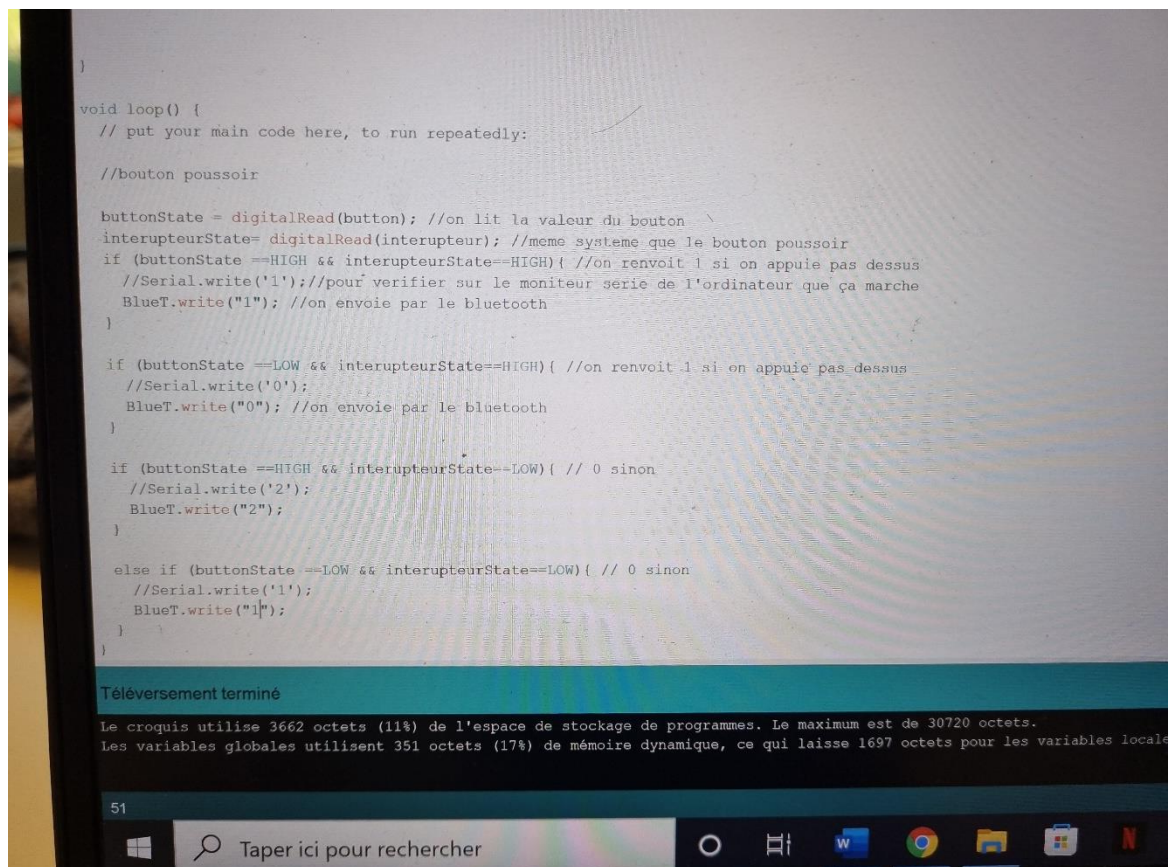
CLIGNOSECURE

Rapport de Séance

Lors de cette séance mon binôme et moi-même sommes passés à l'oral, l'oral intermédiaire du projet. Nous étions dernier sur la liste donc pendant que les autres passés nous pouvions avancer sur notre projet. Pendant que mon binôme essayer de comprendre le fonctionnement d'un système que Mr Masson nous a donné pour mettre sur le guidon du vélo et contrôler a distance le bandeaux lumineux. Moi j'essayais de programmer un 2^{ème} bouton qui ferait clignoter la led.

Programmation du 2^{ème} bouton

Lors de la dernière séance j'avais réussi à programmer un bouton poussoir sur le maître afin qu'il allume lorsqu'on appuie dessus la led connecté sur le module esclave (HC-06). Sachant que notre système comportera 3 boutons j'ai commencé à programmer un 2^{ème} bouton qui fera office de feu de détresse. J'ai rencontré plusieurs problèmes liés au fait que le maître envoyait les informations de 2 actions différentes en simultanées. C'est-à-dire que le maître envoyait en simultanée l'action exercée sur le bouton 1 et sur le bouton 2. J'ai longtemps bloqué la dessus jusqu'à ce que je face plusieurs « if » dans le maître de sorte a ce que si le bouton 1 n'est pas pressé et le 2 non plus, tout cela renvoie une seule et même information et j'ai donc fait 4 cas :



```
void loop() {  
  // put your main code here, to run repeatedly:  
  
  //bouton poussoir  
  
  boutonState = digitalRead(button); //on lit la valeur du bouton  
  interrupteurState= digitalRead(interrupteur); //meme systeme que le bouton poussoir  
  if (buttonState ==HIGH && interrupteurState==HIGH){ //on renvoie 1 si on appuie pas dessus  
    //Serial.write('1');//pour verifier sur le moniteur serie de l'ordinateur que ça marche  
    BlueT.write("1"); //on envoie par le bluetooth  
  }  
  
  if (buttonState ==LOW && interrupteurState==HIGH){ //on renvoie 1 si on appuie pas dessus  
    //Serial.write('0');  
    BlueT.write("0"); //on envoie par le bluetooth  
  }  
  
  if (buttonState ==HIGH && interrupteurState==LOW){ // 0 sinon  
    //Serial.write('2');  
    BlueT.write("2");  
  }  
  
  else if (buttonState ==LOW && interrupteurState==LOW){ // 0 sinon  
    //Serial.write('1');  
    BlueT.write("1");  
  }  
}
```

Téléversement terminé

Le croquis utilise 3662 octets (11%) de l'espace de stockage de programmes. Le maximum est de 30720 octets.
Les variables globales utilisent 351 octets (17%) de mémoire dynamique, ce qui laisse 1697 octets pour les variables locales

51

Taper ici pour rechercher

J'ai décidé pour l'instant de faire en sorte que si les 2 sont appuyés en même temps cela éteins la led mais cela est provisoire et nous verrons par la suite avec mon binôme ce que nous ferons sur le bandeaux lumineux.

Du cote de l'esclave j'ai donc fait en sorte d'allumer la led si on appuie sur le bouton 1 et de la faire clignoter si on appuie sur le bouton 2 et voila le résultat :

<https://youtu.be/zHGT2TZvS3s>

Le clignotement de la led m'a posé pas mal de problème car je ne pouvais pas utiliser la fonction « delay » que nous avons beaucoup utilisé jusqu'ici. En effet cette fonction arrêta le programme et ne faisait même pas clignoter la led donc ce n'était que des problèmes. Mr Masson m'a donc conseillé d'utiliser la fonction « millis() » que l'on avait étudié en cours. Cela m'a demandé beaucoup de temps car j'avais du mal à comprendre son fonctionnement mais après quelques recherches sur internet le problème était résolu.

Prochaine séance :

Pour la prochaine séance, il va falloir que je me penche encore plus sur l'utilisation de la fonction « millis() » car mon binôme n'a mis que des « delay » dans son programme. L'objectif sera donc de corriger son programme et pouvoir commencer le couplage du bandeaux lumineux aux modules bluetooth.