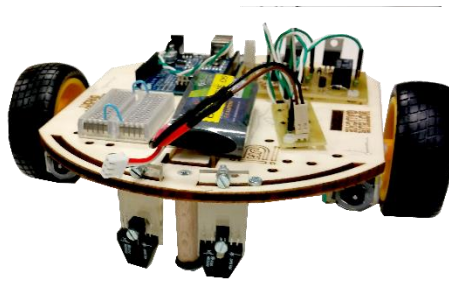




IUT de St Etienne
B.U.T. GEII

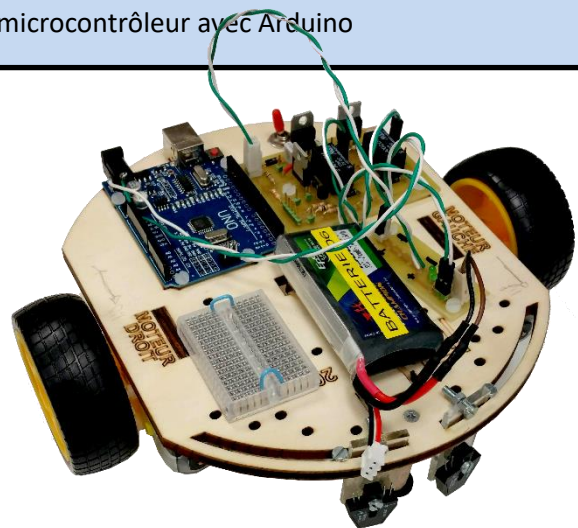
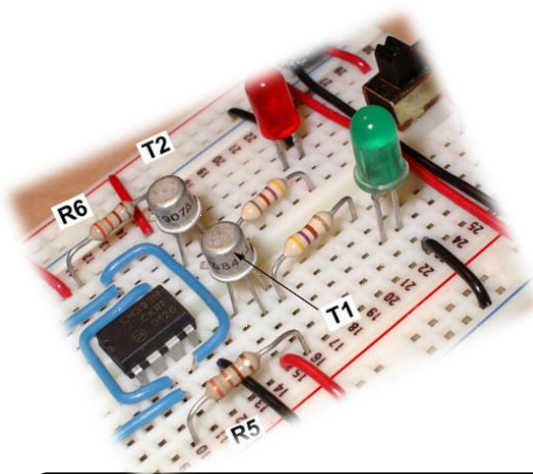


S.A.É. 1

(Situations d'Apprentissage et d'Évaluation)

Phase 07

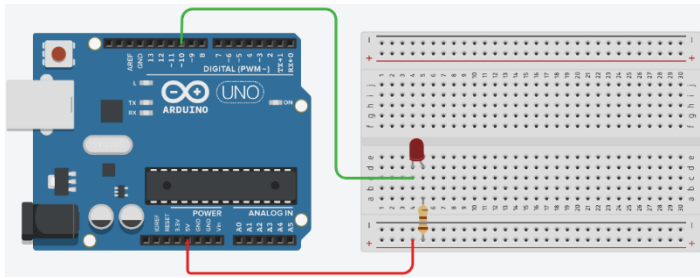
Initiation programmation sur microcontrôleur avec Arduino



1 - Câblage de la carte Arduino

Utilisation d'une plaque de connexion (breadboard) : [BREADBOARD](#)

Câblez la résistance et la LED à la carte Arduino comme ci-dessous :



Dessiner le schéma structurel (simple) représentant le câblage ci-contre avec: $+V_{CC}$ (+5V), la led, la résistance, la pin 10 représentée par un label (\rightarrow PIN 10), la masse (GND).

Quel doit être l'état de PIN 10 pour que la led soit allumée? Éteinte?

Raccordez la carte Arduino à votre PC à l'aide

d'un câble USB.

Lancez le logiciel Arduino :

L'environnement de développement Arduino se présente sous la forme suivante :

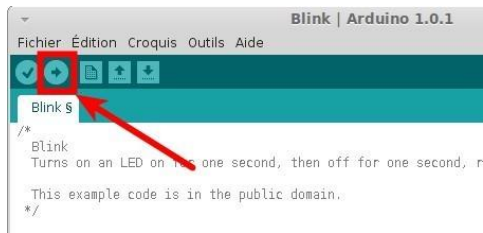
```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly  
}
```

Supprimer tout le texte et saisir le code suivant dans l'éditeur de texte :

```
/* IUT Jean Monnet Département GEII  
Programme de test  
*/  
  
#define PIN_LED 10 // équivalence  
  
void setup() // Fonction exécutée une seule fois, généralement, les initialisations et configurations  
{  
  pinMode(PIN_LED,OUTPUT); // Configuration de la broche PIN_LED_ROUGE en sortie  
}  
  
void loop() // Boucle principale du programme, boucle infinie  
{  
  digitalWrite(PIN_LED, LOW);  
}
```

2 - Etude du programme

Transférez le programme dans l'Arduino à l'aide de la commande Téléverser



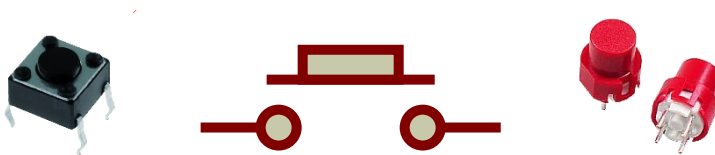
2.a Que constatez-vous ?

2.b Que faut-il modifier dans le programme pour que la LED s'éteigne ?

2.c Testez cette modification.

2.d Modifiez le programme afin de faire clignoter la LED avec une période de 1s.

3 - Utilisation d'un bouton poussoir



On désire maintenant se servir du bouton poussoir afin de commander l'allumage de la LED. Pour cela, on branchera le bouton poussoir sur une entrée **logique** de l'Arduino.

On rappelle qu'une entrée logique ne doit pas être laissée à un état indéterminé.

Une résistance de rappel permet de fixer une entrée numérique à un état haut (« 1 » logique) ou bas (« 0 » logique) stable.

Il existe deux types de résistance de rappel, un type pour chaque état:

- «pull-up»: résistance de rappel vers le niveau haut (+V_{CC})
- «pull-down»: résistance de rappel vers le niveau bas (GND, masse)

3.a Proposez un schéma de câblage pour le bouton poussoir sur l'Arduino.



Validation Prof

3.b Modifiez l'en-tête du programme en définissant une équivalence **BP** désignant la broche à laquelle est raccordé le bouton poussoir.

3.c Modifiez le contenu de la fonction **SETUP** afin de définir cette broche comme une entrée.

3.d Cherchez l'instruction permettant de lire l'état d'une entrée digitale de la carte Arduino.

3.e Donner l'algorithme en pseudo-code (français) afin que la LED s'allume lorsqu'on appuie sur le bouton poussoir

3.f Coder le programme et tester.

3.g Donner l'algorithme en pseudo-code (français) de façon à obtenir le résultat suivant:

Bouton poussoir relâché: la led clignote à 0.5 Hz

Bouton poussoir appuyé: la led clignote à 5Hz

3.h Coder le programme et tester.

3.i Donner l'algorithme en pseudo-code (français) de façon à obtenir le résultat suivant:

Au démarrage la led clignote à 1Hz

1^{er} clic (appui bref) sur le bouton poussoir: la led clignote à 3Hz.

2^{ème} clic sur le bouton poussoir: la led clignote à 10Hz.

3^{ème} clic sur le bouton poussoir: la led clignote à nouveau 1Hz.

Etc.

3.j Coder le programme et tester.

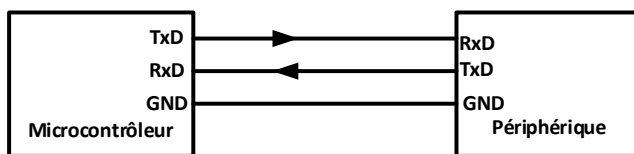
4 - Piloter plusieurs leds en tout ou rien;

On met à votre disposition une plaque de connexion ([Breadboard](#)), prenez 4 leds dans le tiroir à composants et 4 résistances de 330Ω.

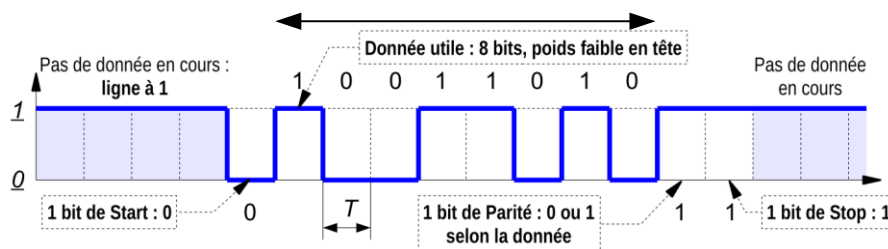
- Faites un schéma de façon à piloter les leds sur les sorties 3, 5, 6 et 9 de l'Arduino.
- Écrire un programme qui permet de faire un chenillard. (une seule led allumée à la fois qui défile 3 ⇒ 5 ⇒ 6 ⇒ 9 ⇒ 3 ⇒ 5 ⇒ 6 etc.)

5 - Utilisation du port série pour communiquer

Le microcontrôleur de l'Arduino (ATMEGA328) dispose d'un port série, c'est-à-dire 1 entrée (RxD) et une sortie (TxD) qui permettent au microcontrôleur de communiquer par des données numériques envoyées en série (1 bit après l'autre par paquets de 8 bits) avec un autre périphérique (un PC, un autre Arduino, un capteur etc.).



Format des données:



Le bit de parité est optionnel, il permet de vérifier une éventuelle erreur dans la communication.

T est la durée d'envoi d'un bit. Le débit est donc $D = \frac{1}{T}$. On l'exprime en bits/s.

Le protocole de communication doit-être connu à priori sinon la communication ne se fait pas.

Donc les paramètres à configurer sont:

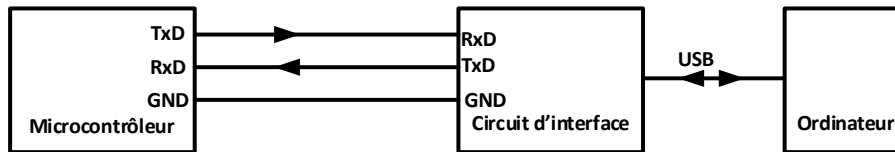
Débit (D), nombre de bits dans la données (de 5 à 8), type de **parité** (paire – Even –a, impaire – Odd – ou pas de parité –N (None)–), nombre de bits de **stop**.

Exemple: 115200 b/s, 8E1 :

Débit 115200 bits/s, 8 bits de données, parité paire, 1 bit de stop.

Cette configuration est à appliquer sur le microcontrôleur et sur le périphérique.

Sur la carte Arduino entre le microcontrôleur et le PC, il y a un circuit qui fait interface entre le port série et l'USB.



Sur l'ordinateur le port série s'appelle COMx (x allant de 1 à 99).

Sur Arduino, le port série est un "objet" intitulé **Serial**.

pour pouvoir communiquer, il faut activer et configurer le port série par :

```
Serial.begin(vitesse,paramètres);
```

Sur le PC, on utilise un logiciel appelé Terminal ou Moniteur. Dans le logiciel Arduino, il y a un moniteur intégré.

5.a Recherche des paramètres de "begin"

Dans la référence Arduino, recherchez les paramètres de la méthode "begin".

5.b Affichage d'un texte

Créer un programme, qui affiche au démarrage du programme, une seule fois: "Bonjour, nous sommes le jj/mois/année".

5.c Affichage d'une variable.

Compléter le programme pour qu'il affiche le contenu d'un compteur toutes les 0.3s.

5.d Interaction

On peut aussi recevoir des données du PC.

Créer un programme qui lit et renvoie tout texte qui lui est envoyé.

6 - Piloter un afficheur 7 segments.

On met à votre disposition une plaque de connexion ([Breadboard](#)), un afficheur à 7 leds appelées segments.

6.a Donner le schéma interne de l'afficheur 7 segments qui vous est fourni.

6.b On veut afficher les chiffres de 0 à 9.

Créer un programme pour afficher en boucle toutes les 0.5s les chiffres de 0 à 9.

6.c On veut afficher un chiffre tapé sur le clavier du PC

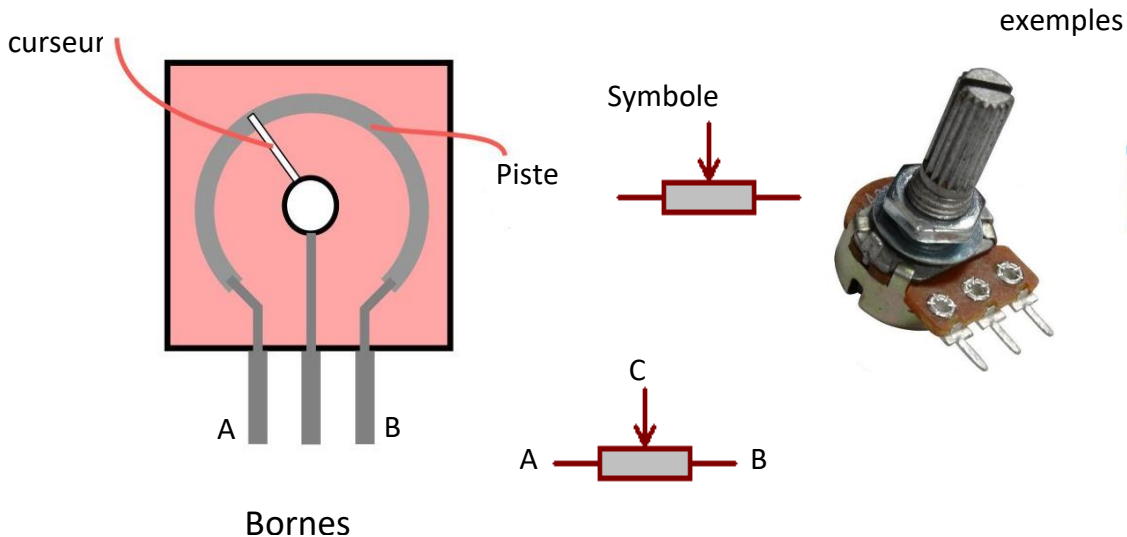
Modifier le programme pour afficher un chiffre que vous tapez sur le clavier du PC et qui est transmis en série à l'Arduino.

7 - Gestion des entrées/sorties analogiques

Vous allez maintenant utiliser une entrée analogique sur laquelle vous ferez varier la tension d'entrée de 0 à 5 V à l'aide d'un potentiomètre.

Un potentiomètre est une résistance variable à trois bornes, la borne centrale est reliée à un curseur se déplaçant sur une piste résistante terminée par les deux autres limites.

Schéma interne du potentiomètre :



On rappelle qu'une entrée ne doit pas être laissée à un état indéterminé.

7.a Proposez un schéma de câblage du potentiomètre, de façon à ce que la tension entre le curseur (C) et la masse varie de 0 à 5V.



Validation prof

Pour pouvoir
analogique du

Un **CAN**
physique (tension, courant)

C

être traitée par le microcontrôleur, la tension présente sur une entrée
microcontrôleur doit être numérisée.

(convertisseur analogique/numérique) transforme une grandeur
physique (tension, courant) en une valeur numérique (binaire).

7.b Câblez le schéma de la question précédente validé par l'enseignant et écrivez un programme permettant de lire la tension présente sur l'entrée analogique et de visualiser le résultat de la conversion analogique/numérique dans le moniteur série.

Identifiez l'instruction de programmation à utiliser.

7.c Donnez le résultat de la conversion analogique numérique lorsque la tension d'entrée vaut 0 V, 1 V, puis lorsqu'elle vaut 5 V.

En déduire sur combien de bits s'effectue la conversion.

7.d En vous inspirant des exercices précédents, réalisez un clignotant à vitesse variable. La fréquence de clignotement de la LED est commandée par la position du curseur du potentiomètre (fréquence de clignotement la plus lente pour 0 V et pour 5 V la plus rapide).