

Nom :

OBJECTIF : PREMIÈRE PROGRAMMATION DE LA CARTE

LE MATÉRIEL

- Une carte UNO + environnement de programmation arduino
- Une carte de prototypage et deux fils
- Deux résistances : 200 ohm et une un peu plus grande
- Une LED

PRÉSENTATION DE L'ARDUINO UNO

Arduino, et son récent synonyme Genuino, est une marque qui couvre des cartes matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due). Les schémas de ces cartes sont publiés en licence libre. Cependant, certains composants, comme le microcontrôleur par exemple, ne sont pas sous licence libre.

Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, de l'informatique embarquée, etc.

C'est une plate-forme basée sur une interface entrée/sortie simple. Il était destiné à l'origine principalement mais pas exclusivement à la programmation multimédia interactive en vue de spectacle ou d'animations artistiques, ce qui explique en partie la descendance de son environnement de développement de Processing, lui-même inspiré de l'environnement de programmation Wiring (l'un pensé pour la production d'applications impliquant des graphismes et l'autre pour pilotage de salles de spectacles).

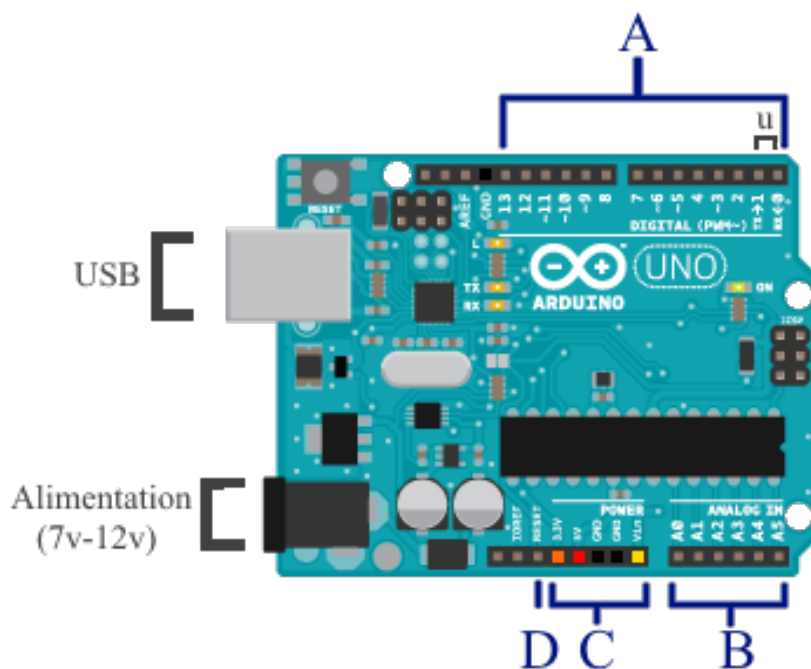


Figure 1: Principe des cartes arduino

Les différentes versions des Arduino fonctionnent sous le même principe général :

A : ce sont les broches dites numériques (0 ou 1) ou " tout ou rien " ; elles offrent en sortie du 5 V et acceptent en entrée du 5 V sur le même principe.

- fonctions digitalWrite() et digitalRead() et pour les ports PWM analogWrite().

B : ce sont les broches dites analogiques, valeur entre 0 V et 5 V

- fonction analogRead()

C : les différentes broches d'alimentation :

-sortie 5 V (+)

-sortie 3,3 V (+)

-les masses (-)

-entrée reliée à l'alimentation (7 V-12 V)

Plus de détail sur les entrées sorties et leur programmation se trouve dans le très utile "PINOUT" montré sur la figure 2.

FAMILIARISATION AVEC LES SPÉCIFICATIONS TECHNIQUES DE LA CARTE UNO ET DE SON MICROCONTRÔLEUR ATMEGA 328P

Pour cette partie, une recherche sur internet d'informations pertinente est nécessaire.¹

1 Quelle est la tension de fonctionnement du microcontrôleur ?

2 Quelle est la tension recommandée par le port d'alimentation noté en figure 1 ?

3 Quel courant peut fournir une pin IO²³ ?

4 Combien y'a t'il de pin numériques ?

5 Combien y'a t'il de pin analogiques ?

6 Quels protocoles de communications entre composants le ATMEGA 328P permet-il ?

LA CARTE ARDUINO UNO : FAIRE CLIGNOTER LA LED 13

7 Installer l'environnement de développement arduino depuis www.arduino.cc. Ouvrir le programme blink en le cherchant dans les menus. Sauvegarder une copie du programme blink sous un autre nom.

8 Si vous avez une carte UNO, nous devons peut-être installer un driver sur l'ordinateur ; c'est le cas si le composant qui fait l'adaptation entre l'USB (connecté au PC) et le port série du microcontrôleur est un CH340G. Dans les cartes originales, ce composant peut aussi bien être un atmel, un FTDI. Dans les cartes peu chères, un CH340G très peu cher est utilisé et nécessite l'installation d'un driver. Dans la figure 1, ce composant se localise entre le port USB et les deux LED Tx et Rx. Sur la carte qui est à votre disposition, identifier le nom de ce composant. Identifier le nom de ce composant dans

¹Un industriel, une encyclopédie, de la documentation technique, etc...

²pin se dit broche en français

³IO signifie In Out, entrée sortie en français

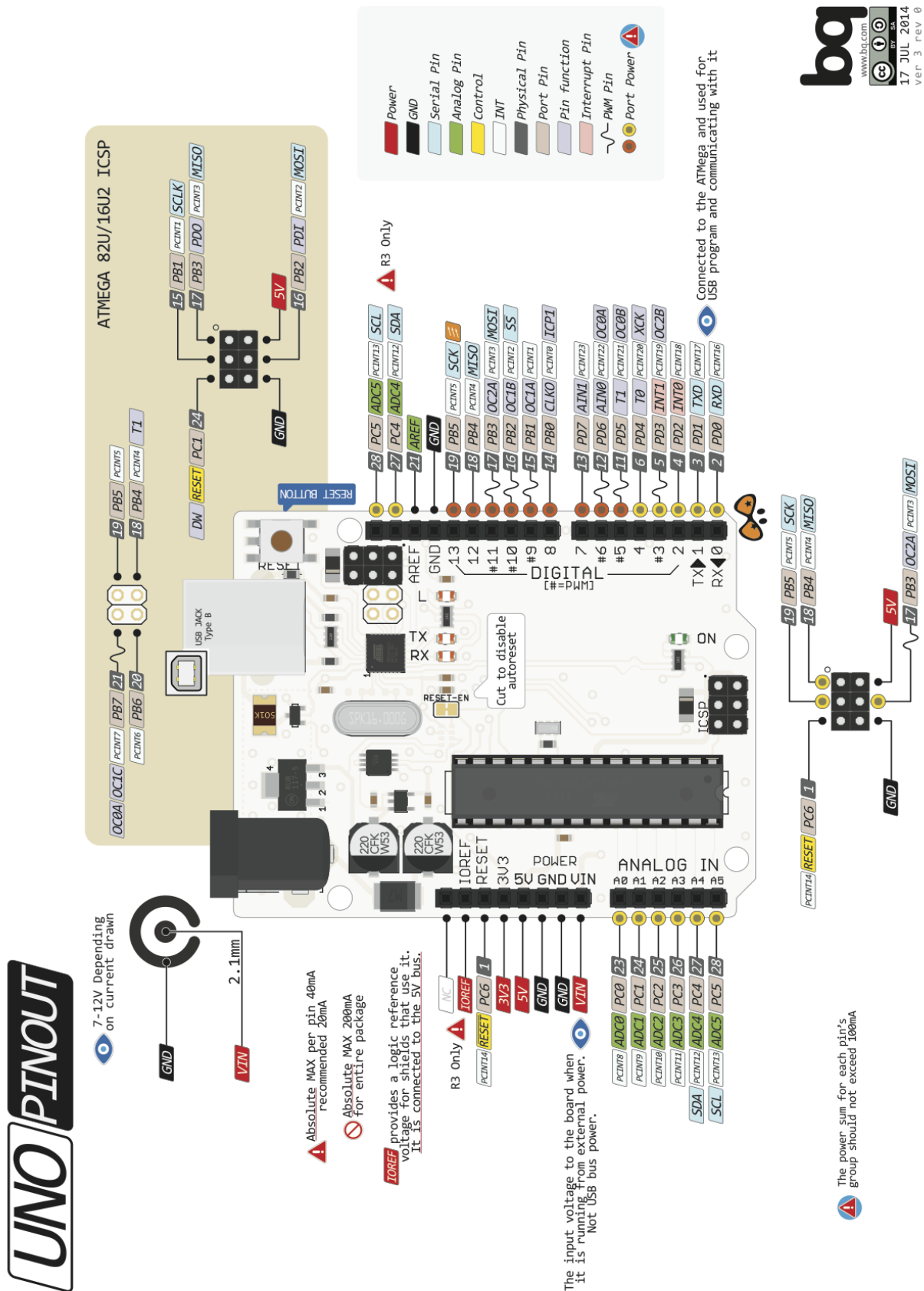


Figure 2: Digramme des entrées-sorties.

votre système d'exploitation.

9 Si c'est un CH340, précisez la procédure d'installation du driver.

.....

.....

Code source du programme Blink

```
1 // the setup function runs once when you press reset or power the board
2 void setup() {
3   // initialize digital pin 13 as an output.
4   pinMode(13, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
10  delay(1000);              // wait for a second
11  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
12  delay(1000);              // wait for a second
13 }
```

10 Téléverser le programme blink. Il faut configurer deux paramètres dans les menus : lesquels ?

.....

11 Est-ce que ce qui suit "/" est exécuté ?

.....

12 A quoi servent ces lignes ?

.....

13 Que signifie en français le commentaire // the setup function runs once when you press reset or power the board ?

.....

14 Que signifie en français le commentaire // the loop function runs over and over again forever ?

15 Pourquoi peut t'on l'appeler la LED L la LED de la pin 13 ?

.....

16 A quoi servent les LED TX et RX ? A quelles pins sont elles branchées ?

.....

.....

17 Quelle est la période de clignotement de la LED dans le programme blink tel qu'il est dans le menu exemples ?

.....

18 Combien de temps s'écoule entre deux allumage de la LED de la pin 13 ?

.....

19) Quelle est donc la période de clignotement de la LED de la pin 13 ?

20) Quelle est donc sa fréquence ?

21) Pour un clignotement à 1 Hz, la période est de 1s. Quels paramètres changer pour l'implémenter ? Préciser son nom et sa valeur.

22) De même pour 10 Hz.

23) Faire clignoter la LED à 0,5 Hz, et vérifier avec un voltmètre que la broche de sortie de la carte UNO passe périodiquement de 0V et 5V.

24) Faire clignoter la LED à 0,5 Hz, et vérifier avec un voltmètre que la broche du microcontrôleur correspondante est périodiquement 0V ou 5V.

25) Faire clignoter la LED à 0,5 Hz, et vérifier avec un voltmètre que la tension aux bornes de la LED est d'environ 2V (ça dépend surtout de la couleur de la LED).

Bilan :

- le fichier blink sauvegardé sous un autre nom ;
- le port série utilisé est identifié dans le gestionnaire de périphériques de l'ordinateur ;
- la LED clignote à 0,5Hz ;
- le voltmètre mesure bien 0V-5V sur une pin du microcontrôleur, une pin de la carte ;
- le voltmètre mesure bien 2V aux bornes de la LED 13.

Constatation professeur :

CONNECTER UNE LED EXTERNE

Code source #1:

```
1 int ledPin = 10; // define digital pin 10.
2 void setup() {
3   pinMode(ledPin, OUTPUT); // define pin with LED connected as output.
4 }
5 void loop() {
6   digitalWrite(ledPin, HIGH); // set the LED on.
7   delay(1000); // wait for a second.
8   digitalWrite(ledPin, LOW); // set the LED off.
9   delay(1000); // wait for a second
10 }
```

26) Dans les conditions de ce montage, une valeur classique pour la résistance de polarisation de la LED est 220 ohm. Normalement ça dépend de la couleur de la LED, mais comme le microcontrôleur et la carte sont bien protégés des erreurs, on se permet de faire à peu près. Quel est le calibre optimal pour la mesurer avec un multimètre ?

27) Pourquoi ne convient-il pas d'utiliser un calibre en dessous ?

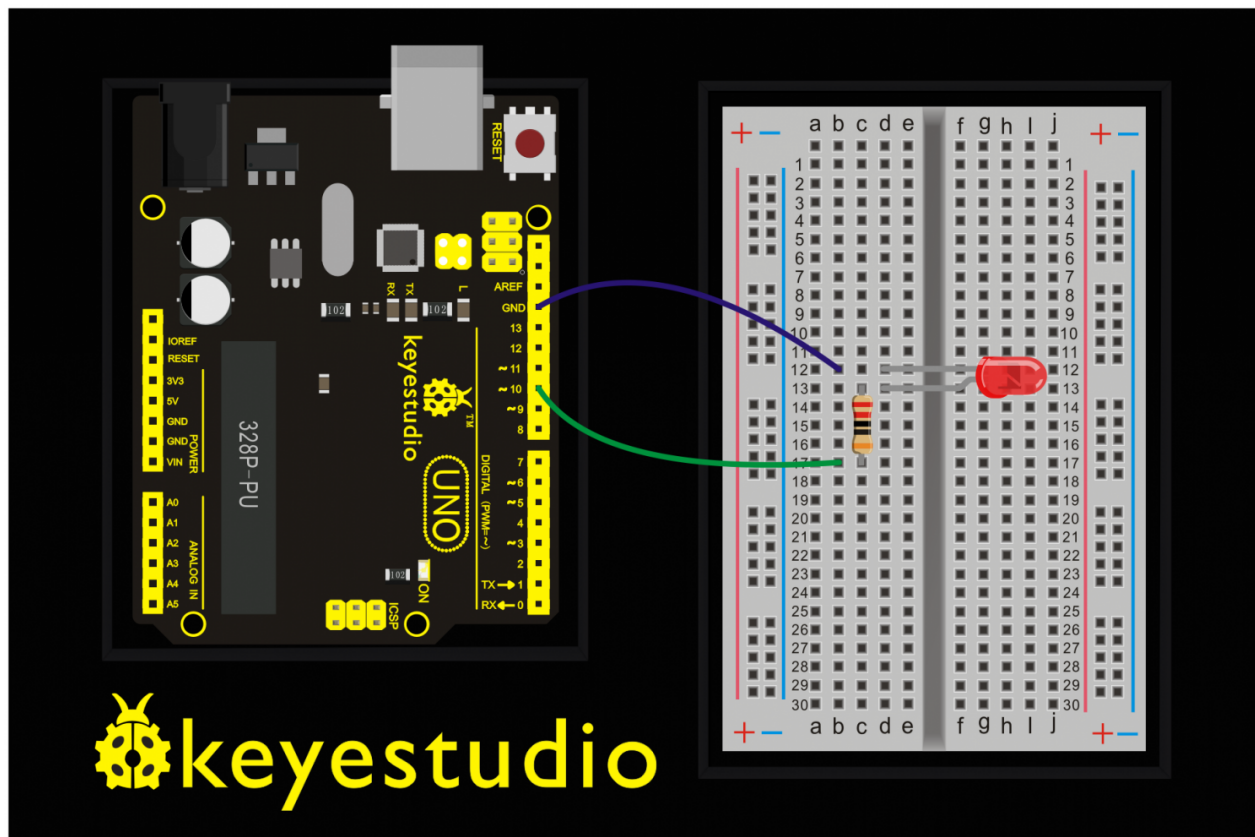


Figure 3: Cablage d'une LED externe

28 Pourquoi ne convient-il pas d'utiliser un calibre en dessus ?

29 Avec votre multimètre, mesurer la valeur exacte de la résistance que vous choisissez ?

30 Implémentez le code Code source #1.

31 Utilisez une valeur de resistance un peu plus grande. Qu'observez-vous ?

32 Pour les deux résistances, saurez-vous calculer et mesurer les courants qui circulent dans la LED ? Précisez vos calculs et vos mesures.

Bilan:

- resistance mesurée sur le bon calibre ;
- le montage marche ;
- observation avec une résistance plus grande.

Constatation professeur :