

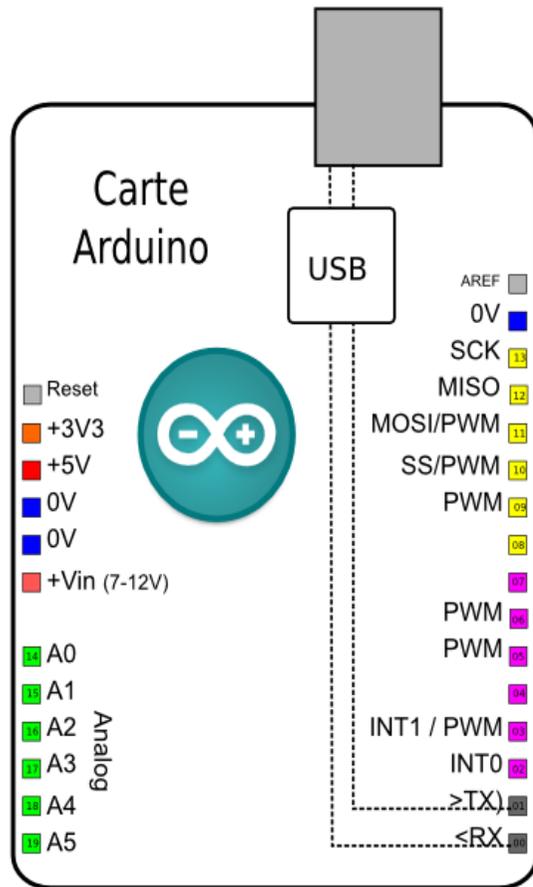
| Langage Arduino | Description / Syntaxe | Exemple |
|---|--|---|
| setup() | Fonction d'initialisation des variables, du sens des broches, des bibliothèques utilisées. Contient le code qui n'a besoin d'être exécuté qu'une seule fois au démarrage. | <pre>void setup() { Serial.begin(9600); pinMode (A5,OUTPUT); } void loop() { int E=analogRead(A0); //Conversion analogique-numé //R est branchée à la borne //int E renvoie une valeur e U=E*5.0/1024; //Calcul de la tension analo //E*5/1024 ne convient pas c //Pour obtenir une valeur dé //Pour cela, il faut écrire temperature=(U-1.269)/0.046; //Calcul de la tempé Serial.print("Température = "); Serial.print(temperature); Serial.println("°C"); delay(2000); }</pre> |
| loop() | Fonction exécutée en boucle sans fin, permettant à un programme de s'exécuter et de répondre. | |
| <i>Les fonctions setup() et loop() sont obligatoires même si elles ne contiennent aucun code.</i> | | |
| void | Obligatoire devant les fonctions setup() et loop(). Indique un type de fonction. Le mot Void veut dire "vide" et signifie qu'une fonction de ce type ne renvoie aucune valeur | |
| { } | Les accolades sont les "conteneurs" du code du programme. Elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades. | |
| // | Utilisés pour ajouter des commentaires et comprendre le programme. Les lignes qui commencent par // sont ignorées par le compilateur. | |
| ; | Les points virgules sont obligatoires et terminent les instructions. | |
| pinMode() | Permet de configurer une broche pour qu'elle se comporte en entrée (haute impédance) ou en sortie. pinMode(broche, mode) <ul style="list-style-type: none"> broche : le numéro de la broche de la carte Arduino. mode : soit INPUT (entrée en anglais) (ou 0) ou OUTPUT (sortie en anglais) (ou 1) | <p>pinMode (3,INPUT) configure la broche 3 en entrée.</p> <p>pinMode (12,OUTPUT) configure la broche 12 en sortie.</p> <p>pinMode (15,OUTPUT) ou pinMode (A2,OUTPUT) configure la broche A2 en sortie.</p> |
| digitalWrite() | Met un niveau logique HIGH (HAUT) ou LOW (BAS) sur une broche numérique. Si la broche a été configurée en SORTIE avec l'instruction pinMode(), sa tension de sortie est à 5 V pour le niveau HAUT et à 0 V pour le niveau BAS. digitalWrite(broche, valeur) <ul style="list-style-type: none"> broche: le numéro de la broche de la carte Arduino valeur : HIGH (ou 0) ou LOW (ou 1) | <p>digitalWrite(13, HIGH) met la broche 13 au niveau haut.</p> <p>digitalWrite(10, LOW) met la broche 10 au niveau bas.</p> |
| int | Déclare une variable de type int (pour integer, entier en anglais). int var = val <ul style="list-style-type: none"> var : le nom de la variable val : le numéro de la broche <p>Les variables de type int sont utilisées pour le stockage de nombres entiers, sur 2 octets (16 bits), de -32 768 à +32 767</p> | <p>int Valeur_Capteur = 0 déclare une variable de type int appelée Valeur_Capteur et ayant pour valeur 0.</p> <p>int Pin_Capteur = A0 déclare une variable de type int appelée Pin_Capteur ayant pour valeur 14 (correspond au numéro de la broche A0). Cette déclaration permet de sélectionner la broche A0 en la désignant par Pin_Capteur ultérieurement dans le programme.</p> <p>int Pin_LED = 13 déclare une variable de type int appelée Pin_LED ayant pour valeur 13. Cette déclaration permet de sélectionner la broche 13 en la désignant par Pin_LED ultérieurement dans le programme.</p> |
| long | Déclare des variables de type long. long var = val <ul style="list-style-type: none"> var : le nom de la variable | long T déclare une variable de type long appelée T. |

| | | |
|--------------------------|---|--|
| | <ul style="list-style-type: none"> val : le numéro de la broche <p><i>Les variables de type long sont des variables de taille élargie pour le stockage de nombres entiers, sur 4 octets (32 bits), de -2 147 483 648 à +2 147 483 647.</i></p> | |
| float | <p>Déclare des variables de type "virgule-flottante".</p> <p>float var = valeur</p> <ul style="list-style-type: none"> var : le nom de la variable val : la valeur de la variable <p><i>Les variables de type float sont des variables de taille élargie pour le stockage de nombres à virgule, sur 4 octets (32 bits), de -3,4028235×10⁺³⁸ à +3,4028235×10⁺³⁸.</i></p> | <p>float pi = 3.1415927 déclare une variable de type float appelée pi et ayant pour valeur 3,1415927.</p> <p>float z déclare une variable de type float appelée z.</p> |
| const | <p>Le mot const signifie constante. C'est un qualificateur de variable qui modifie le comportement de la variable, en en faisant une variable "lecture seule". Ceci signifie que la variable peut être utilisée comme n'importe quelle autre variable du même type, mais sa valeur ne peut pas être changée dans le programme (autrement dit, elle est initialisée une fois pour toute lors de sa déclaration). On obtient une erreur de compilation si on essaye de modifier la valeur d'une variable const.</p> | <p>const float pi = 3.14; // définit la constante pi</p> |
| analogRead() | <p>Convertit la tension analogique présente sur la broche spécifiée en une valeur numérique entière comprise entre 0 et 1023.</p> <p>analogRead(broche)</p> <ul style="list-style-type: none"> broche : le numéro de la broche analogique de la carte Arduino <p><i>La carte Arduino comporte 6 entrées (8 voies sur la Mini et la Nano) connectées à un convertisseur analogique-numérique 10 bits. Cela signifie qu'il est possible de transformer la tension d'entrée entre 0 et 5 V en une valeur numérique entière comprise entre 0 et 1023. Il en résulte une résolution (écart entre 2 mesures) de : 5 V / 1024 intervalles, autrement dit une précision de 0,0049 V. Une conversion analogique-numérique dure environ 100 µs pour convertir l'entrée analogique. Par défaut, ces broches mesurent entre le 0 V (valeur 0) et le 5 V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant l'instruction analogReference().</i></p> | <p>analogRead(A0) ; //convertit la tension analogique présente sur la broche A0 en une valeur numérique.</p> |
| analogReference() | <p>Configure la tension de référence utilisée avec les entrées analogiques. La fonction analogRead() renverra la valeur 1023 pour une entrée analogique égale à la tension de référence.</p> <p>analogReference(type)</p> <ul style="list-style-type: none"> type : DEFAULT: la valeur par défaut de la tension de référence analogique = 5 V. INTERNAL : une tension de référence interne, égale à 1,1 V sur l'ATmega 168, et à 2,56 volts sur l'ATmega 8. EXTERNAL : la tension de référence appliquée sur la broche AREF utilisée en tant que référence en tension. | <p>analogReference (INTERNAL) modifie la tension de référence à 1,1 V (au lieu de 5V par défaut lors d'une conversion analogique numérique.</p> |
| Serial.begin() | <p>Fixe le débit de communication en bits par seconde (bps) pour la communication série entre la carte Arduino et le PC.</p> <p>Serial.begin(debit)</p> <ul style="list-style-type: none"> debit : débit de communication en pratique fixé à 9600 | <p>void setup() { Serial.begin(9600); // ouvre le port série et fixe le débit de communication à 9600 bp }</p> |

| | | |
|----------------------------|--|---|
| Serial.print() | <p>Affiche les données sur le port série.</p> <p>Serial.print(val) Serial.print(val, format)</p> <ul style="list-style-type: none"> val : la valeur à afficher. N'importe quel type de données. format : spécifie la base utilisée (pour les nombres entiers) ou le nombre de décimales (pour les nombres de type float) | <p>Serial.print(78); // affiche "78" Serial.print(1.23456); // affiche "1.23" Serial.print(1.23456,3); // affiche "1.235" Serial.print(byte(78)); // affiche "N" (dont la valeur ASCII est 78) Serial.print('N'); // affiche "N" Serial.print("Hello world."); // affiche "Hello world."</p> |
| Serial.println() | <p>Affiche les données sur le port série suivi d'un retour de chariot et d'un saut de ligne. Cette instruction a la même forme que l'instruction Serial.print().</p> <p>Serial.println(val) Serial.println(val, format)</p> <ul style="list-style-type: none"> val : la valeur à afficher. N'importe quel type de données. format : spécifie la base utilisée (pour les nombres entiers) ou le nombre de décimales (pour les nombres de type float) | <p>Serial.println(78); // affiche "78" et passe à la ligne.</p> |
| pulseIn() | <p>Lit la durée d'une impulsion (soit niveau HAUT, soit niveau BAS) appliquée sur une broche configurée en ENTREE.</p> <p>Par exemple, si le paramètre valeur est HAUT, l'instruction pulseIn() attend que la broche passe à HAUT, commence alors le chronométrage, attend que la broche repasse au niveau BAS et stoppe alors le chronométrage.</p> <p>L'instruction renvoie la durée de l'impulsion en microsecondes. L'instruction s'arrête et renvoie 0 si aucune impulsion n'est survenue dans un temps spécifié.</p> <p>pulseIn(broche, valeur)</p> <ul style="list-style-type: none"> broche : le numéro de la broche sur laquelle il faut lire la durée de l'impulsion. (type int) valeur : le type d'impulsion à lire : HIGH (niveau HAUT) ou LOW (niveau BAS). (type int) | <p>duree = pulseIn(3,High); // mesure la durée de l'impulsion de niveau haut dans la variable duree.</p> |
| delay() | <p>Réalise une pause dans l'exécution du programme pour la durée (en milliseconde) indiquée en paramètre.</p> <p>delay(ms)</p> <ul style="list-style-type: none"> ms : le nombre de millisecondes que dure la pause. | <p>delay(1000); // met le programme en pause pendant 1 s.</p> |
| delayMicroseconds() | <p>Réalise une pause dans l'exécution du programme pour la durée (en microseconde) indiquée en paramètre.</p> <p>delayMicroseconds(us)</p> <ul style="list-style-type: none"> us : le nombre de microsecondes que dure la pause. | <p>delayMicroseconds(5); // met le programme en pause pendant 5µs.</p> |

Les variables : Les accents sont interdits pour nommer les variables. Il est intéressant de nommer les variables en utilisant des majuscules pour les différencier dans le programme

Le langage Python est utilisé pour réaliser les tests, les comparaisons et les boucles.



Les broches numériques sont numérotées de 2 à 13. Ces broches fonctionnent en 5 V. Chacune peut fournir ou recevoir une intensité maximum de 40mA.

Les broches numériques repérées par PWN ou ~ peuvent être utilisées en sorties pour fournir un signal dit PWM (Pulse Width Modulation ou modulation de largeur d'impulsion).

Les broches analogiques sont désignées par les repères A0 à A5 et sont numérotés de 14 à 19 Elles sont également utilisables en tant que broches numériques.

La broche 13 est connectée à une LED repérée par la lettre L sur la carte. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La broche AREF permet d'appliquer une tension de référence externe.

¹ L'ensemble des broches de la carte Arduino ne peut fournir que 200 mA. Par conséquent, il faut limiter l'intensité utilisée pour chaque broche à une dizaine de mA avec des résistances adaptées : 270 Ω pour une LED par exemple (ce qui fait environ 13 mA sur la broche).