

Le test conditionnel IF

Le principe :

SI Test vrai

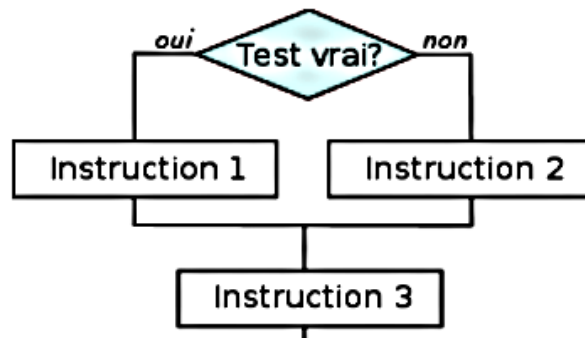
Instruction 1 exécutée

SINON

Instruction 2 exécutée

FIN SI

Instruction 3 (suite du programme)



Extrait de la documentation Arduino:

if (conditional) and ==, !=, <, > (comparison operators)

if, which is used in conjunction with a comparison operator, tests whether a certain condition has been reached, such as an input being above a certain number. The format for an if test is:

```
if (someVariable > 50)
{
  // do something here
}
```

The program tests to see if someVariable is greater than 50. If it is, the program takes a particular action. Put another way, if the statement in parentheses is true, the statements inside the brackets are run. If not, the program skips over the code.

The brackets may be omitted after an if statement. If this is done, the next line (defined by the semicolon) becomes the only conditional statement.

```
if (x > 120) digitalWrite(LEDpin, HIGH);
```

```
if (x > 120) digitalWrite(LEDpin, HIGH);
```

```
if (x > 120){ digitalWrite(LEDpin, HIGH); }
```

```
if (x > 120){
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
} // all are correct
```

The statements being evaluated inside the parentheses require the use of one or more operators:

Comparison Operators:

$x == y$ (x is equal to y)

$x != y$ (x is not equal to y)

$x < y$ (x is less than y)

$x > y$ (x is greater than y)

$x \leq y$ (x is less than or equal to y)

$x \geq y$ (x is greater than or equal to y)

if / else

if/else allows greater control over the flow of code than the basic if statement, by allowing multiple tests to be grouped together. For example, an analog input could be tested and one action taken if the input was less than 500, and another action taken if the input was 500 or greater. The code would look like this:

```
if (pinFiveInput < 500)
{
    // action A
}
else
{
    // action B
}
....
```

En résumé cela donne :

```
if (.....) // debut du test
{
    instruction 1
}
else
{
    instruction 2
} // fin du test
```

les "....." représentent une équation logique.
Exemples :

- $a > b$
- $a \geq b$
- $a < b$
- $a \leq b$
- $a == b$ (noter les deux = sinon a prend la valeur de b)
- $a != b$ (si a différent de b)

Application : appui sur un bouton poussoir

Objectif du programme : la led s'allume si on appuie sur un bouton poussoir (quand on relâche elle s'éteint)

Principe:

- On va rajouter un bouton poussoir que l'on va raccorder sur l'entrée numérique (digital) n° 2.
- On va utiliser un test conditionnel IF

Le montage complet

Le schéma de raccordement :

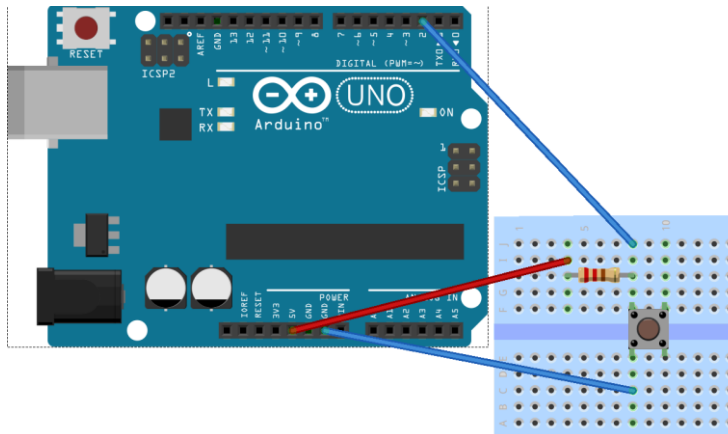
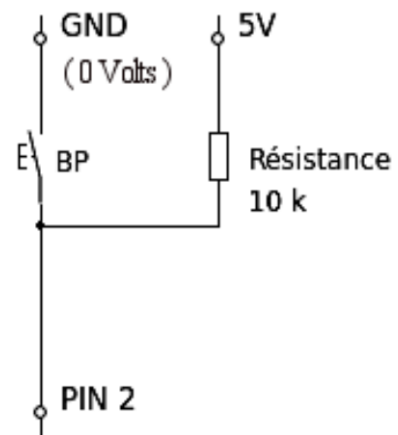


Schéma électronique :



A quoi sert la résistance de 10kΩ?



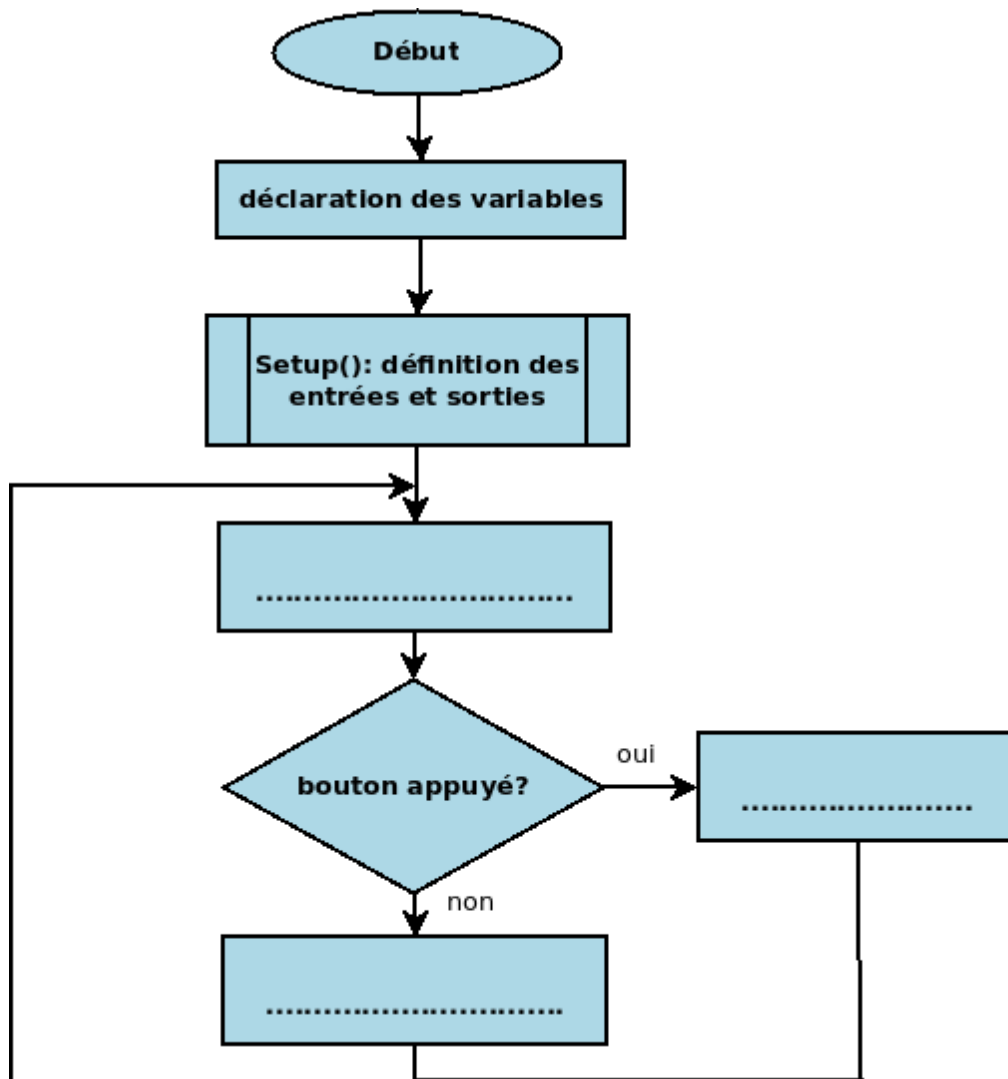
Lorsqu'une broche numérique d'Entrée n'est pas connectée, le niveau présent sur cette broche d'entrée est instable et la LED dans notre programme clignoterait de façon aléatoire. Ceci est dû au fait que l'entrée est "flottante", c'est à dire qu'elle n'a pas de connexion matérielle au +5V ou à la masse, et elle oscille de façon aléatoire entre le niveau HAUT et le niveau BAS. C'est pour cette raison qu'il faut mettre une résistance de "rappel au plus" (appelée aussi résistance de tirage ou pull-up) dans le montage du BP sur la broche numérique en entrée.

1. Faire le petit câblage (le arduino doit être débranché du cordon USB) et faites vérifier par le professeur.

Le programme Arduino

2. Ouvrir le fichier « clign_led_BP.txt », récupérer le programme contenu dans le fichier et testez le.
3. Compléter l'algorithme du programme :

S1 : le test « IF »



4. Que constatez-vous ? L'objectif de fonctionnement est-il atteint ? Si ce n'est pas le cas expliquez ce qui se passe (à partir du schéma électronique).
5. Proposez et essayez une modification du programme. Faire valider par le professeur.



Le microcontrôleur utilisé sur la carte Arduino dispose d'une résistance interne qu'il est possible d'activer pour réaliser le Pull-up (rappel au plus), ce qui permet de se dispenser d'une résistance externe !

Dans le programme Arduino cela donne:

Le seul changement se situe dans le Setup():

```
pinMode(boutonPin, INPUT); //met la broche en entree  
digitalWrite(boutonPin, HIGH) ; // activation du pullup  
de la broche en entrée
```

