MOOC Programmer un objet avec Arduino
- Capteurs numériques -
***
Baptiste GAULTIER
Ingénieur de Recherche à IMT-Atlantique

Salut à tous.

Je suis ravi de vous retrouver pour une nouvelle vidéo où on va parler de capteurs, des composants électroniques simples qui vont permettre à Arduino d'interagir avec son environnement.
Ces capteurs permettent de mesurer de nombreuses grandeurs physiques, la luminosité, la température, le niveau de bruit ou encore la distance par rapport à un objet.

Allez, on commence avec un montage simple, un Arduino qui va allumer une LED lorsqu'on appuie sur un bouton. Pour réaliser ce montage, nous aurons besoin d'un Arduino, d'une platine de prototypage, un bouton-poussoir, une LED, des fils de prototypage et des résistances de 1 kiloohm. Voici le montage à réaliser. Je vous invite à mettre la vidéo en pause le temps de réaliser ce montage, avec le simulateur ou avec Arduino.

Passons maintenant au code. Comme d'habitude, nous n'allons pas taper grand-chose. Je vous invite à ouvrir l'exemple Button ou à copier le code placé en dessous de la vidéo, vous devriez avoir ceci sous les yeux. Détaillons maintenant les nouvelles instructions présentes dans ce programme Button.
Après quelques lignes de commentaires, nous déclarons une constante de type entier ou int appelé buttonPin, qui aura pour valeur 2. Notez le mot-clé const qui signifie Constant. Cela signifie que nous ne souhaitons pas que cette variable puisse être modifiée par la suite.
Ensuite, on déclare une variable appelée buttonState et on l'initialise à zéro. Il n'y a pas de mot-clé const qui précède la déclaration, ce qui signifie que la valeur de buttonState pourra être changée durant l'exécution du programme.

On passe ensuite au bloc setup qui compte les instructions qui doivent être exécutées une seule fois, au début. On va retrouver la fonction pinMode. Elle permet de spécifier si une brochette est une entrée ou une sortie. On vient ici spécifier que le pin ledPin est une sortie, du courant va sortir pour alimenter la LED, d'où le second paramètre placé à output, tandis que la broche buttonPin va recevoir du courant.

On vient de dire que c'est une entrée, d'où le second paramètre placé à input. C'est tout pour le setup. On passe au bloc loop avec une nouvelle instruction.
Ça se corse un peu. Pour bien comprendre cette ligne, nous devons la lire par la fin.
DigitalRead est une fonction qui va lire la valeur d'une broche. Ici, on vient lire la valeur de la broche 2. Le résultat de cette fonction va ensuite être placé dans buttonState.

À chaque tour de boucle loop, la variable buttonState, que nous avons déclarée au début du programme, va donc prendre une valeur high ou low, haut ou bas en anglais, selon si on appuie sur le bouton branché sur la broche 2. Nous venons ensuite tester cette valeur grâce aux lignes suivantes.
Dans un premier temps le mot if suivi de parenthèses permet de tester si la condition buttonState vaut high est vraie. En d'autres termes, est-ce qu'on a du courant sur le port 2.
Notez la présence de deux = consécutifs. Il ne faut surtout pas oublier de mettre un second égal pour faire un test, sinon on fait une initialisation.

Si on a du courant reçu sur le port 2, alors on passe à ce qui se trouve dans le bloc if, en l'occurrence je viens allumer la LED, sinon je passe à ce qui se trouve dans le bloc else. En l'occurrence, j'éteins la LED. On s'arrête là pour le loop.

Ces instructions vont être exécutées encore et encore, tant que l'Arduino est branché. Si on résume, dans la loop, nous venons lire la broche 2.
Si on reçoit du courant, c'est que le bouton est appuyé, donc on allume la LED, sinon on l'éteint. Nous pouvons maintenant téléverser ou uploader notre programme sur Arduino et vérifier que le programme fonctionne.

On s'arrête ici avec cette vidéo. Merci de l'avoir regardée et à bientôt.