|  |
| --- |
| **Annexe 1 : « Programme en langage Arduino (C/C++) »** |

// COMMANDER UN BUZZER PIEZZOELECTRIQUE AVEC UN INTERUPTEUR A GLISSIERE // fonctions de base arduino: <http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Reference> // L'objectif de ce montage est de commander un buzzer avec un interrupteur à glissière de la manière suivante: // Lorsque l'interrupteur à glissière est fermé, le buzzer bipe 3 fois // Lorsque l'interrupteur à glissière est ouvert, le buzzer ne bipe pas

// Définition des broches numériques utilisées du microcontrôleur. // Remarque : on utilise la commande "const int" pour les nombres entiers

const int Buzzer = 10; // Broche 10 utilisée pour le buzzer piezoélectrique - Elle se nomme maintenant : "Buzzer" const int IG = 8; // Broche 8 utilisé pour l'interrupteur à glissière - Elle se nomme maintenant : "IG"

// BOUCLE D'INITIALISATION : exécutée une seule fois // Définition des broches numériques d'entrée et de sortie du microcontrôleur

void setup() { // Contenu de l'initialisation Serial.begin(9600); // ouvre le port série et fixe le débit de communication à 9600 bauds (nombre de symboles transmissibles par seconde) pinMode(Buzzer, OUTPUT); // "Buzzer" est une broche de sortie pinMode(IG, INPUT); // "IG" est une broche d'entrée }

// BOUCLE PRINCIPALE : exécutée à l'infini

void loop() { // Contenu du programme int InterGlissiere = digitalRead(IG); // Lecture de l'entrée numérique IG et stockage du résultats dans InterGlissiere if(InterGlissiere==HIGH) // Si "InterGlissiere1" est à l'état HAUT (l'interrupteur est fermé) { for (int i=0; i<3; i++) // Boucle qui commande le nombre de bips à exécuter (i allant de 0 à 2) { digitalWrite(Buzzer,HIGH); // Met un niveau logique HAUT sur la broche numérique "Buzzer"; La tension de la broche est de 5V - Le buzzer est alimenté tone (10, 1000, 100); // Le buzzer (broche numérique 10) bipe à 1000 Hz pendant 100ms delay(300); // Pause de 300 ms entre chaque bip } delay(3000); // Pause pendant 3000 ms Serial.println("Le buzzer bipe 3 fois"); } else // Sinon { digitalWrite(Buzzer, LOW); // Met un niveau logique BAS sur la broche numérique "Buzzer"; La tension de la broche est mise à 0V - Le buzzer n'est pas alimenté delay(2000); // Pause pendant 2000 ms Serial.println("Le buzzer ne bipe pas"); } }