Formation Arduino

Première chose

Connection de la carte sur l’ordinateur et chargement d’un programme de test

Description des différents zone de programmation

Appel aux bibliothèques

Configuration des constantes et des variable

Avant le setup nous pouvons créer des constances et définir les différents variable de programmation et permet de stocker des valeurs

Exemple :

const int buttonPin = 2;

const int ledPin = 13;

int buttonState = 0;

Setup

Dans le setup nous allons configurer le mode de fonctionnement des interfaces de l’arduino , initialisation

Cette section n’est exécuté qu’une seule fois lors du démarrage de l’arduino

Exemple :

void setup()

{

  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT);

pinMode(2, INPUT\_PULLUP);

 // Configuration de la connexion série

   Serial.begin(9600);

}

Puis nous avons le programme

Exemple :

void loop() {

 buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {

    digitalWrite(ledPin, HIGH);

  }

else {

    digitalWrite(ledPin, LOW);

Serial.println("Pas de Détection");

  }

Les instructions

**Règle de base**

Toutes instructions se termine par un ;

Les systèmes effectue une différence entre minuscule et majuscule

Commentaire // ou /\* pour plusieurs ligne \*/

**Pas d’accents**

Pour les variables les caractères utilisables sont ([a-z], [A-Z], [0-9]) et \_

**Type de variable :**

Int , long,char, float , double, byte, word, boolean…

Taille des différentes zone mémoire

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type |  | Valeur max et min | Nombre de bits | Nombre d’octets |
| int | +/- | -32768 ⬄32767 | 16 | 2 |
| long | +/- | -2147483648 ⬄2147483647 | 32 | 4 |
| char | +/- | -127 ⬄127 | 8 | 1 |
| float | 0.0 | -3.4 10exp38 ⬄ 3,4 10 exp 38 | 32 | 4 |
| Double | 0.0 | -3.4 10exp38 ⬄ 3,4 10 exp 38 | 32 | 4 |
| boolean | + | 0 ⬄1 | 1 | 1 |
| word | + | 0 à 65535 | 16 | 2 |
| byte | + | 0 à 255 | 8 | 1 |
| Unsigned char | + | 0 à 255 | 8 | 1 |
| Unsigned int | + | 0 à 65535 | 16 | 2 |
| Unsigned long | + | 0 ⬄ 4294967295 | 32 | 4 |

**Opération mathématique :**

+ ,-,/,\*,

%(reste de la division)

**Incrémentation et décrémentation**:

Exemple :

I++ ; ou I=I+1 ;

I-- ; ou I=I-1 ;

Bascule avec un booléen

x= !x ;

**Condition**

* == égalité
* < inférieur
* > Supérieur
* < = inférieur ou égal
* >= supérieur ou égal
* != différent

**Structure de condition**

if(condition) // SI

{

}

else // sinon

{

}

Possibilité d’imbrication

if(condition) // SI

{

}

else if ( ) // sinon

{

}

else

{

}

Le switch ( le sélecteur dépendant d’une valeur)

**switch** (options\_voiture)

{

**case 0**:

// il n'y a pas d'options dans la voiture

**break**;

**case 1**:

// la voiture a l'option GPS

**break**;

**case 2**:

// la voiture a l'option climatisation

**break**;

**case 3**:

// la voiture a l'option vitre automatique

**break**;

**case 4**:

// la voiture a l'option barres de toit

**break**;

**case 5**:

// la voiture a l'option siège éjectable

**break**;

**default**:

// retente ta chance ;-)

**break**;

}

**Opérateur Logique**

Permet de combiner plusieurs conditions ensemble

&& ET

|| OU

! Non

**Mot de fin pour la théorie**

Il reste à voir dans le futur la gestion de boucle et de tableau et l’utilisation des fonctions et bibliothèque

## Pratique :

Création d’un programme pour activer une led à l’aide d’un bouton poussoir

Une DEL / LED : Diode Electro-Luminescente, ou bien ”Light Emitting Diode” en anglais



Tension de fonctionnement (Tension maximum direct) (+/- 1,2 à 1,6V) dépend de la fiche technique du composant.

Tension maximum inverse pour une diode classique +/- 75V

Le courant de passage (+/- 20ma)

Rappel loi d’ohm U =R\*I

Calcul de la résistance de la led

Tension led +/- 1,2 V Tension Arduino 5V courant led 0,02A

R = (5-1,2)/0,02 🡺 190 ohms

La planche de test ou breadboard

Exemple de connexion



Vue arrière

Branchement de la led

Tips

Avec le brochage de la carte Arduino, vous devrez connecter la plus grande patte au +5V (broche *5V* ). La plus petite patte étant reliée à la résistance, elle-même reliée à la broche numéro 2 de la carte. Tout ceci a une importance. En effet, on pourrait faire le contraire, brancher la LED vers la masse et l’allumer en fournissant le 5V depuis la broche de signal. Cependant, les composants comme les microcontrôleurs n’aiment pas trop délivrer du courant, ils préfèrent l’absorber.

Pour cela, on préférera donc alimenter la LED en la plaçant au +5V et en mettant la broche de Arduino à la masse pour faire passer le courant.



Programmation

1. Définir la pin de la led

**const int** led\_rouge = **2**; // définition

1. Définir l’utilisation de la pin

pinMode(led\_rouge, OUTPUT);

1. Allumer la led

digitalWrite(led\_rouge, LOW);

Pourquoi LOW ?

Utilisation de la commande delay

* Explication de la fonction millis()

Gestion des boutons

Explication de pull-up et pull-down

Le but est de supprimer le parasitage et d’obtenir un signal précis.

Pull-up



Là nous avons un signal non flottant mais nous risquons d’avoir des rebonds



Pour compenser cela nous allons ajouter un condensateur



Utilisation des pull-ups interne

Cela nous évite de devoir ajouter des résistances

pinMode(Bouton, INPUT\_PULLUP );

Cependant, si vous les mettez en marche, il faut se souvenir que cela équivaut à mettre la broche à l’état haut (et en entrée évidemment).



Programme set et reset (en cours)

Programmation

1. Définir la pin de la led

int const bouton = 2 ;

**int** etat;

1. Définir l’utilisation de la pin

pinMode(bouton, INPUT\_PULLUP );

1. Allumer la led

etat = digitalRead(bouton); // Rappel : bouton = 2

**if**(etat == HIGH){

// le bouton est relaché

**}**

**Else**

**{**

**}**

// le bouton est appuyé

#define

Exemple

#define VERT 0

Le ”#define” est ce que l’on appelle **une directive de préprocesseur**. Lorsque le logiciel

Arduino va compiler votre programme, il va remplacer le terme défini par la valeur qui le suit.

Donc ici VERT correspond à zéro

Cela équivaut dans ce cas à une constante

const int VERT = 0;

Futur

Branchement d’un potentiomètre



Analogique (entrée et sortie) (A0 à A5)

valeur = analogRead(3); // Nous ne devons pas mettre A3 donne valeur de 0 à 1023 🡺 de 0 à 5v par pas de 4,88mv

Convesion via la fonction map()

Exemple

// conversion de la valeur lue en tension en mV

tension = map(valeurLue, **0**, **1023**, **0**, **5000**);

// conversion des mV en V

tension = tension / **1000**;

Au retour de la liaison série (seulement si on envoie les valeurs par la liaison série) on aurait

donc (valeurs à titre d’exemple) :

valeurLue = 458

tension = 2.290V

sortie analogique

PWM

Fonction

Bibliothèque

Exemple #include <Servo.h>

STOP 589