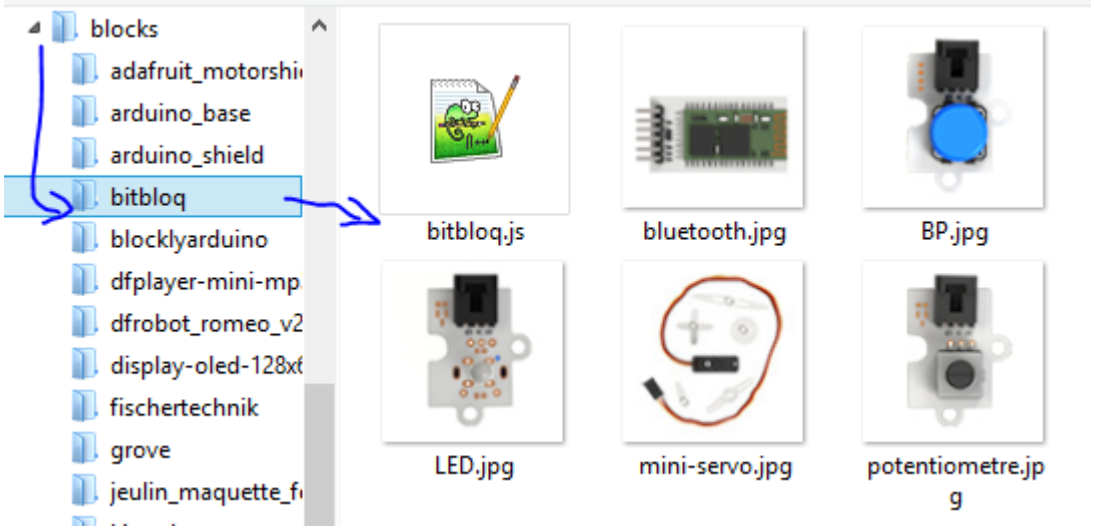
Table des matières

1. Pour créer des blocs, voici la démarche en prenant pour exemple la FlyCamOne. 3

1.1. Le principe : 3

1. Pour créer des blocs, voici la démarche en prenant pour exemple la FlyCamOne.

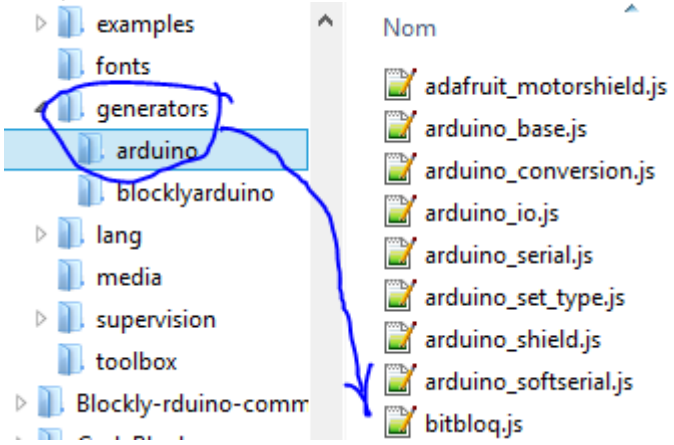
1.1. Le principe :

1. le dossier '**\blocks**' contient un sous-dossier par fabricant afin d'y stocker les images et fichiers de scripts.
1. 

2. un fichier **'\blocks\fabricant\fabricant.js'** va décrire le nom de chaque bloc et de quelle façon Blockly va le dessiner à l'écran. Comme nous allons créer des blocs multilingues, ***aucun texte pour l'utilisateur ne doit être tapé dans ce fichier***, *ils seront dans des fichiers de traduction*.
1.
```
Blockly.Blocks['bq_led'] = {
  init: function() {
    this.setColour(Blockly.Blocks.bq.HUE);
    this.setHelpUrl(Blockly.Msg.BQ_HELPURL);
    this.appendDummyInput()
        .appendField(Blockly.Msg.GROVE_INOUT_LED_INPUT1)
        .appendField(new Blockly.FieldImage(Blockly.pathToBlockly + 'blocks/bitbloq/LED.jpg', Blockly.Arduino.imageSize, Blockly.Arduino.imageSize));
    this.appendValueInput("PIN")
        .setCheck('Number')
        .setAlign(Blockly.ALIGN_RIGHT)
        .appendField(Blockly.Msg.GROVE_INOUT_LED_INPUT2);
    this.setInputsInline(true);
    this.appendDummyInput()
        .setAlign(Blockly.ALIGN_RIGHT)
        .appendField(Blockly.Msg.GROVE_INOUT_LED_INPUT3)
        .appendField(new Blockly.FieldDropdown(Blockly.Msg.FIELDDROPDOWN), 'STAT')
    this.setPreviousStatement(true, null);
    this.setNextStatement(true, null);
    this.setTooltip(Blockly.Msg.BQ_LED1_TOOLTIP);
  }
};
```
2. On lit là tout ce qui constitue le bloc. Enfin on lit surtout qu'on ne lit rien...tout ce qui sera utilisé est défini dans différents fichiers qui contiennent les valeurs des variables comme Blockly.Blocks.bq.HUE, Blockly.Msg.BQ_HELPURL, etc. Toutes ces variables seront expliqués par la suite, donc** il est important de parcourir toute cette documentation pour comprendre la mécanique globale de Google Blockly**. - un fichier **'\blocks\blocks_colors.js'** va contenir toutes les valeurs en hexadecimal des couleurs à utiliser. Par cohérence pour les utilisateurs, il vaut mieux que **la couleur de la catégorie (*définie dans le fichier toolbox expliqué plus bas*) soit la même pour tous les blocs de la catégorie** :

3.
```
Blockly.Blocks.arduino_softserial.HUE = "#00979D";
Blockly.Blocks.arduino_shield.HUE = "#F39800";
Blockly.Blocks.bq.HUE = "#608621";
Blockly.Blocks.BT_ELEC.HUE = "#142D5E";
Blockly.Blocks.dfplayer.HUE = "#F39800";
```

3. un fichier **'\generators\fabricant.js'** va utiliser les éléments décrits dans la définition graphique du bloc (*voir point précédent*) pour enrichir un texte qui sera le code en langage Arduino à produire.

1.


2.
```
Blockly.Arduino.bq_led = function() {
    var dropdown_pin = Blockly.Arduino.valueToCode(this, 'PIN', Blockly.Arduino.ORDER_ATOMIC);
  var dropdown_stat = this.getFieldValue('STAT');
  Blockly.Arduino.setups_['setup_green_led_'+dropdown_pin] = 'pinMode('+dropdown_pin+', OUTPUT);';
  var code = 'digitalWrite('+dropdown_pin+','+dropdown_stat+');\n';
  return code;
};
```

4. le fichier '**\blocks\blocks_typing.js**' va contenir le typage de chacun des blocs que vous avez créé, c'est à dire s'il renvoie une valeur de type entier (int) ou chiffre à virgule (float) ou...

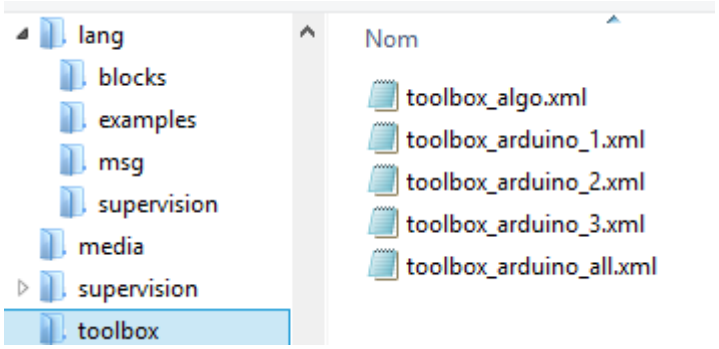
1.
```
//*****************************************************************
//                              Bitbloq
//*****************************************************************

//---------------------------------bitbloq.js-----------------------

Blockly.Blocks.bq_led.getBlockType = function() {
    return Blockly.Types.BOOLEAN;
};
```

5. les fichiers **'\lang\blocks\fr.js' & '\lang\blocs\en.js'** vont contenir toutes les chaînes de caractère que l'utilisateur lira.

1.
```
Blockly.Msg.BQ_PIN = "sur la broche";
Blockly.Msg.BQ_PIN_DIGITAL = "sur la broche Numérique";
Blockly.Msg.BQ_PIN_PWM = "sur la broche PWM~";
Blockly.Msg.BQ_PIN_ANALOG = "sur la broche Analogique";
Blockly.Msg.BQ_HELPURL = "http://www.bq.com/fr/produits/kit-robotica.html";
Blockly.Msg.BQ_LED1_TOOLTIP = "Sortie led (réf : LED)";
```

2.
```
Blockly.Msg.BQ_PIN = "Pin #";
Blockly.Msg.BQ_PIN_DIGITAL = "Digital Pin#";
Blockly.Msg.BQ_PIN_PWM = "PWM~ Pin#";
Blockly.Msg.BQ_PIN_ANALOG = "Analog Pin#";
Blockly.Msg.BQ_HELPURL = "http://www.bq.com/fr/produits/kit-robotica.html";
Blockly.Msg.BQ_LED1_TOOLTIP = "Sortie led (réf : LED)";
```

6. le nom de chacun des blocs que vous voulez afficher dans une catégorie à l'écran devra être ajouté dans le menu ad-hoc, ils sont contenus dans le dossier **'\toolbox\'**.

1. 

2.
```
<category name="CAT_BQ" colour="#608621">
    <category name="CAT_BQ_IN" colour="#608621">
        <block type="bq_ultrason">
            <value name="TRIGER">
                <shadow type="math_number">
                    <field name="NUM">0</field>
                </shadow>
            </value>
            <value name="DIST">
                <shadow type="math_number">
                    <field name="NUM">1</field>
                </shadow>
            </value>
        </block>
        <block type="bq_bouton_poussoir">
            <value name="PIN">
                <shadow type="math_number">
                    <field name="NUM">1</field>
                </shadow>
            </value>
        </block>
```

From:
https://wiki.libreduc.cc/ - **LibrEduc**

Permanent link:
https://wiki.libreduc.cc/fr:arduino:blockly_rduino:creerblocsmultiling:principe

Last update: **2025/01/16 20:24**