

# Mblock 5 pour l'animation STEM

- [Création de blocs d'extensions avec mBlock 5 Extension Builder](#)

# Création de blocs d'extensions avec mBlock 5 Extension Builder

Voir <https://forum.makeblock.com/t/how-to-add-an-extension-for-mblock/2280>

Mblock est normalement installé dans `C: \ProgramData`

Sur mon PC il est installé dans `C: \Users\USER\mblock` (voir aussi dans `C: \Users\Public\Programs`)

Les extensions sont installées dans : `C: \Users\USER\mblock\exts`

Les sources sont en JavaScript... Il faut trouver comment remonter aux sources C (fichiers .c et .h) qui appellent les bibliothèques dont dépendent les blocs de programmation et définissent le code C équivalent à chaque bloc. À défaut, trouver si les fournisseurs de l'extension ont mis le code source sur une forge logicielle publique.

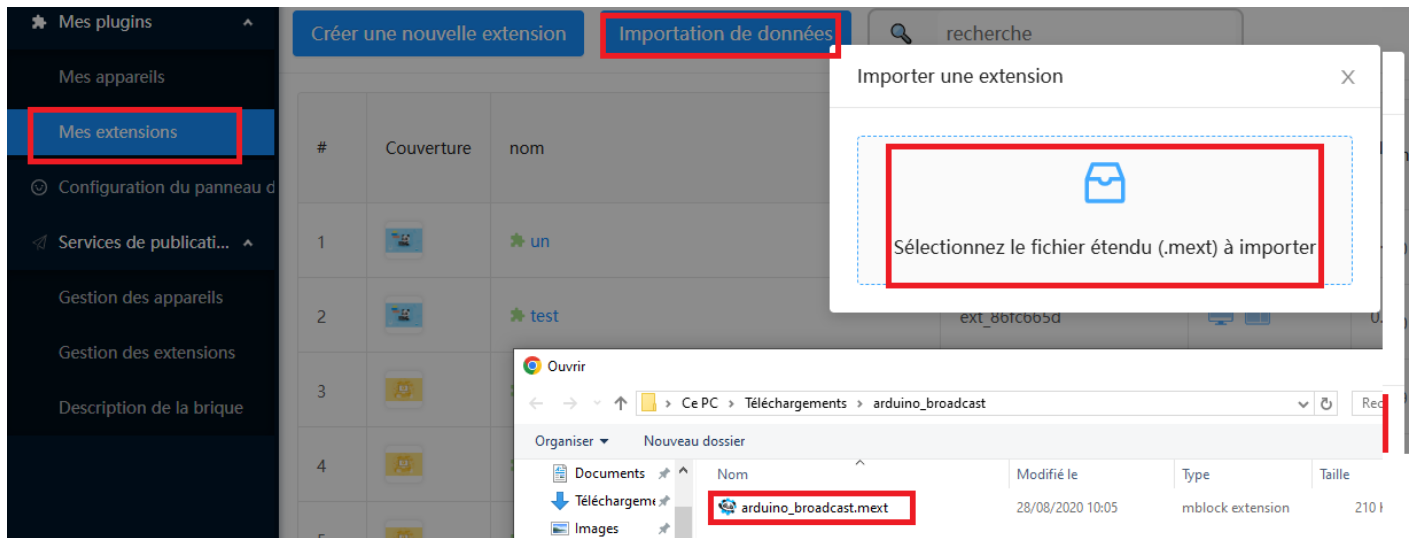
## Création de nouveau bloc avec mBlock 5 Extension Builder

- Créer un compte sur <https://ext.mblock.cc/>
- Une bonne documentation existe en Anglais : <https://www.yuque.com/makeblock-help-center-en/extension-builder>
- Mais elle n'explique pas comment créer un bloc qui fait appel à une bibliothèque externe non disponible dans mBlock. Voir prochain paragraphe

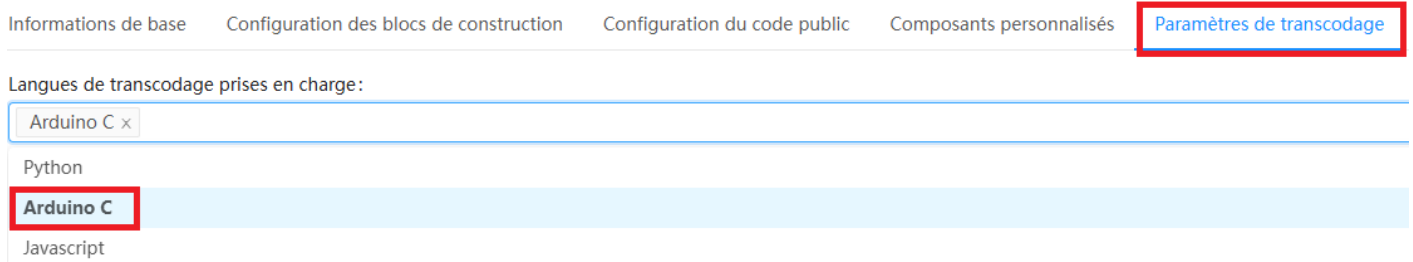
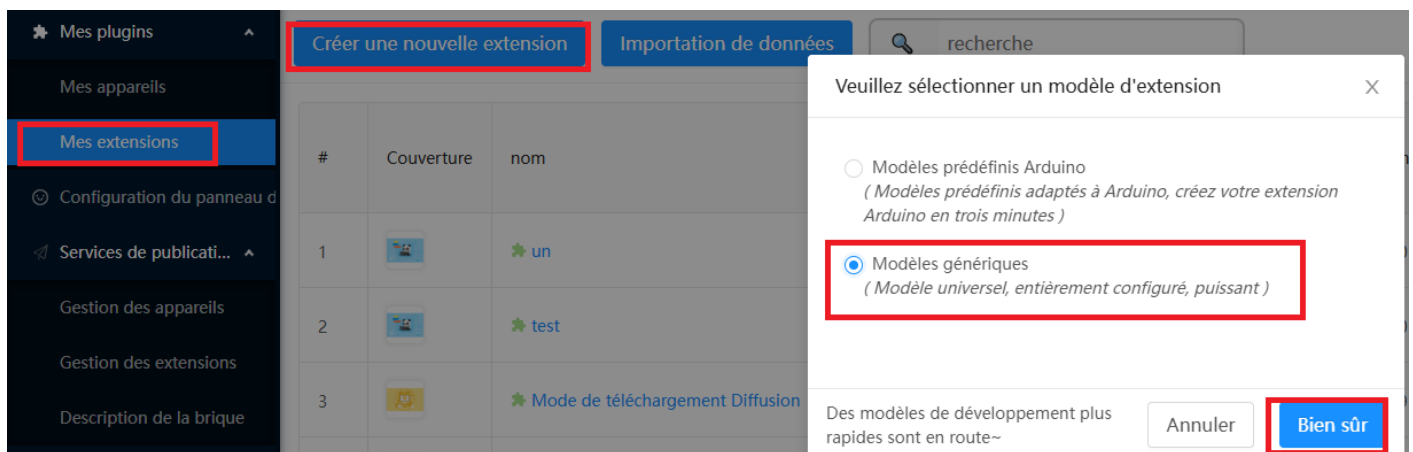
## Avec des bibliothèques de dépendance

<https://support.makeblock.com/hc/en-us/articles/15264543414935-Example-Arduino-Add-a-Third-party-Library-for-an-Arduino-Extension>

Pour inspiration, on importe une librairie existante :



Pour créer une nouvelle librairie :



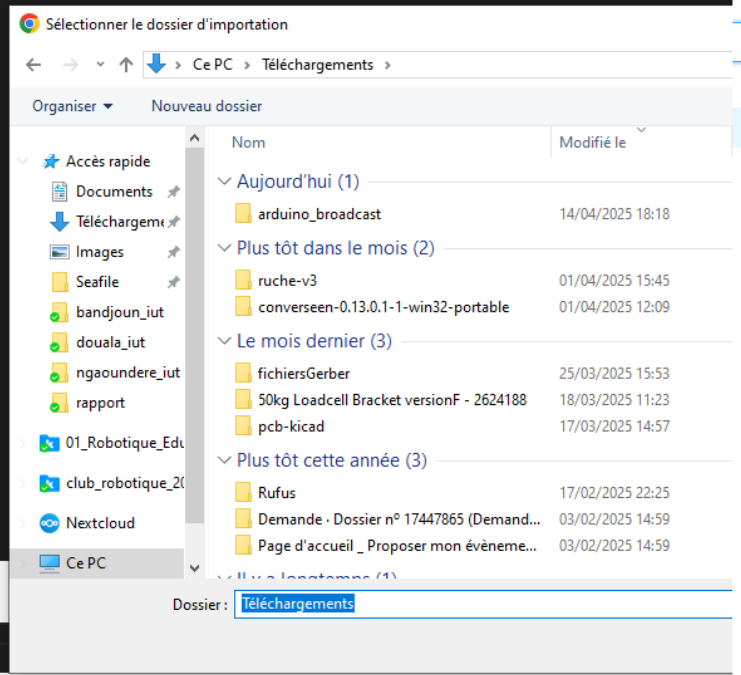
Scroller jusqu'en bas et importer un dossier `src` contenant les fichiers `.h` et `.cpp` de votre lib

```

réducteurs : [
  {
    nom : 'inclure',
    réduire : ( codes ) => {
      laissez codes1 = [ ]
      pour ( laissez code de codes ) {
        laissez codeStr = '' ;
        si ( typeof code === 'chaîne' ) {
          codeStr = code ;
        } else if ( typeof code === 'function' ) {
          codeStr = code ( ) ;
        }
        si ( codes1 . indexOf ( codeStr ) === - 1 ) {
          codes1 . pousser ( codeStr ) ;
        }
      }
      si ( codes1 . longueur === 0 ) {
        retour indéfini ;
      }
      codes de retour1 . carte ( code => {
        retourner '#include ' + code ;
      } ) . rejoindre ( '\n' ) + '\n'
    }
  }
]

```

Configuration de la complétion de code:



Fichiers sources (veuillez placer les fichiers .cpp et .h dans un dossier pour le téléchargement):

## Sources

- Introduction Extension Builder et usage basique : <https://www.yuque.com/makeblock-help-center-en/extension-builder>
- Comment ajouter un bloc qui fait appel à une librairie qui n'est pas dispo dans mBlock ? <https://support.makeblock.com/hc/en-us/articles/15264543414935-Example-Arduino-Add-a-Third-party-Library-for-an-Arduino-Extension>