

**VSM & Arduino UNO – Proteus V8.1**

**TP N°1**

**Ecran LCD et bouton poussoir**



L'objectif de ce TP est de simuler une carte Arduino Uno en association avec un écran LCD 2x16 dans VSM.

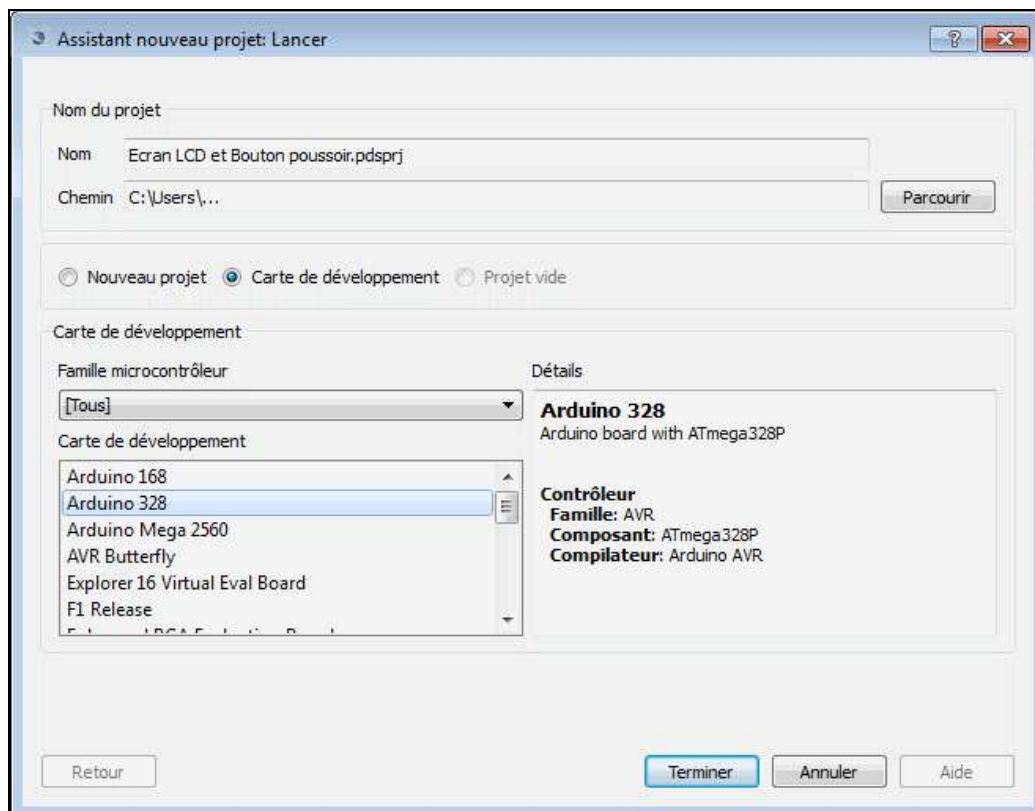
Remarque : Avant toute manipulation, **vérifiez** que vous avez bien téléchargé le compilateur Arduino dans VSM !

- ➔ Pour plus d'informations, nous vous invitons à visionner nos vidéo de démonstration pour l'installation du compilateur Arduino dans VSM et bien d'autres choses encore !

## I. Création du projet

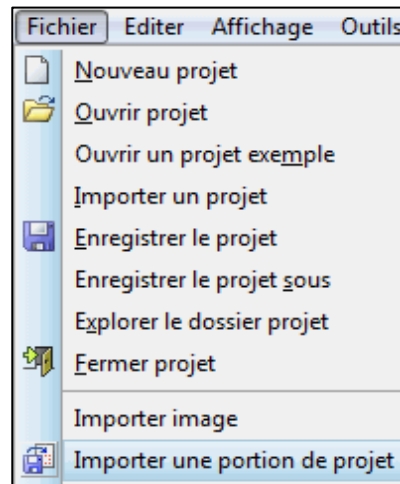
- 1) Ouvrez Proteus et cliquez sur **Nouveau projet**. Une fenêtre apparaît ; indiquez le répertoire où sera enregistré le projet et cochez « Carte de développement ». Sélectionnez Arduino 328 (Arduino UNO), puis donnez un nom à votre projet.

Cliquez sur « Terminer ».

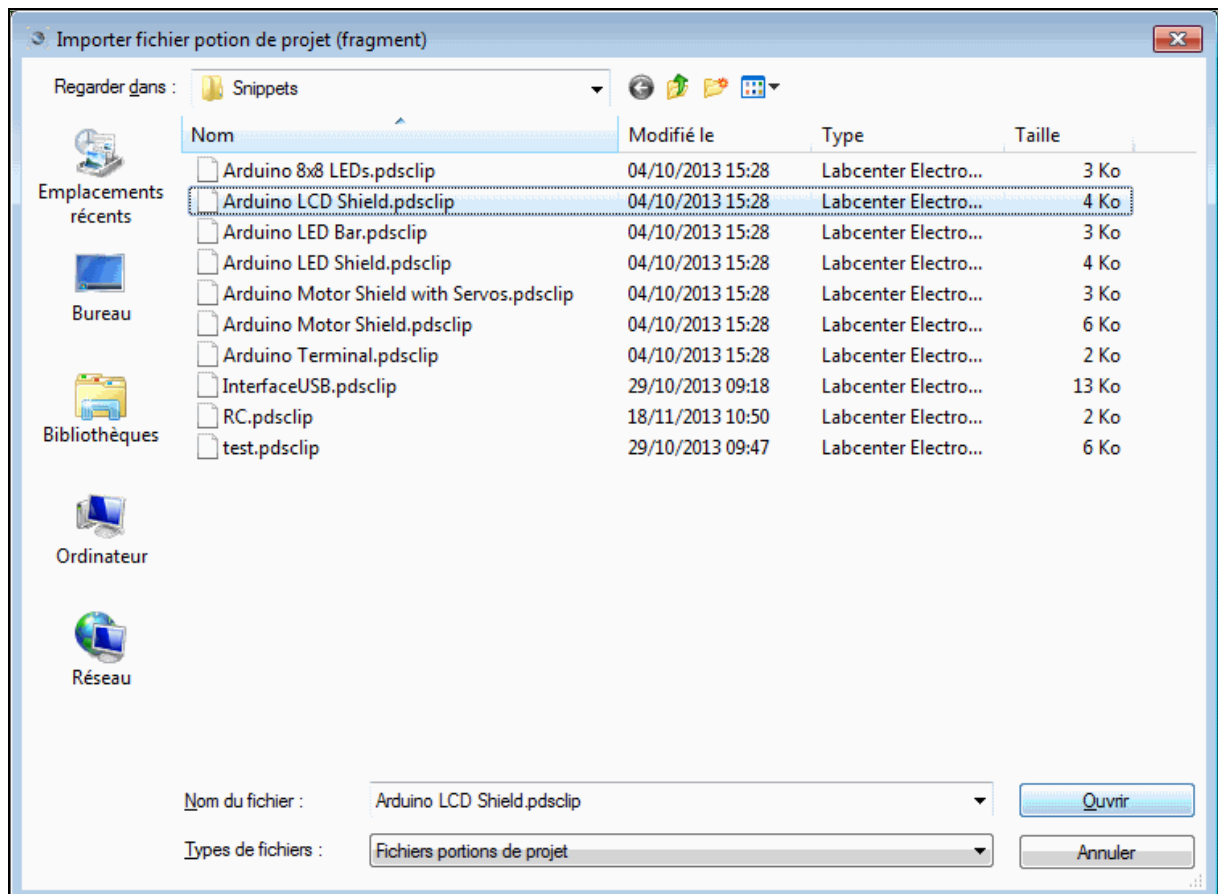


## II. Mise en place des composants

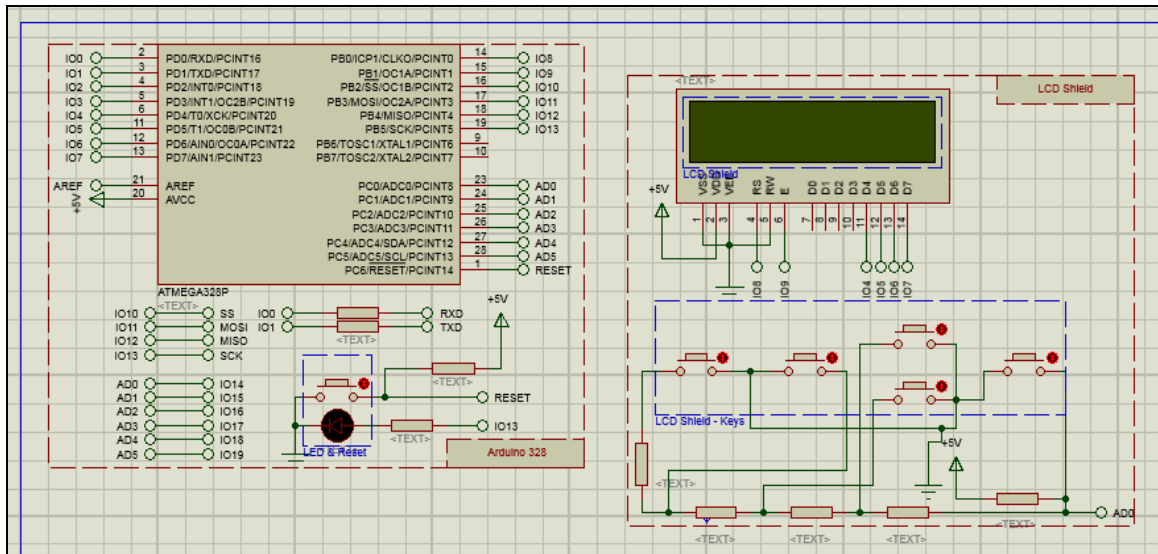
- 1) Proteus ouvre ISIS et VSM Studio. Vous remarquez que la carte est automatiquement ajoutée au schéma. Nous allons inclure manuellement l'écran LCD via le menu « Fichier » puis « Importer une portion de projet ».



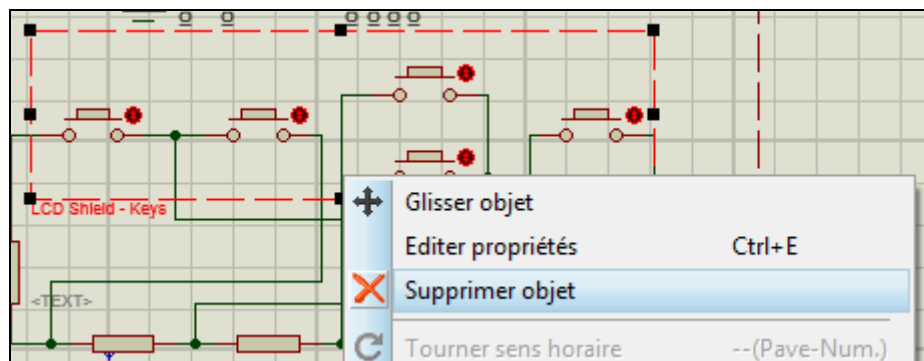
- 2) Dans ce dossier sont enregistrés tous les models pré-câblés de certains modules pour les cartes Arduino. Nous utiliserons « Arduino LCD Shield ». Cliquez sur Ouvrir.



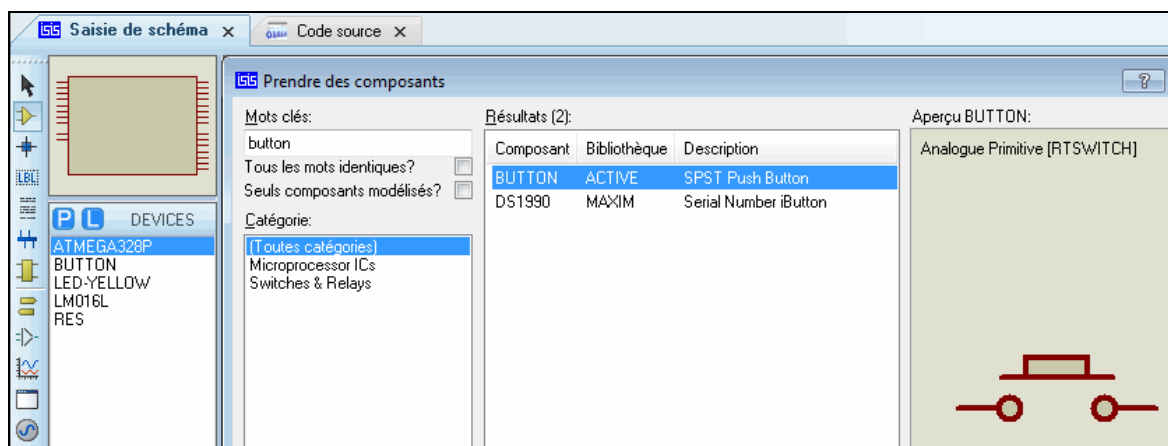
3) Avec un clic gauche, placez l'écran LCD sur le schéma.



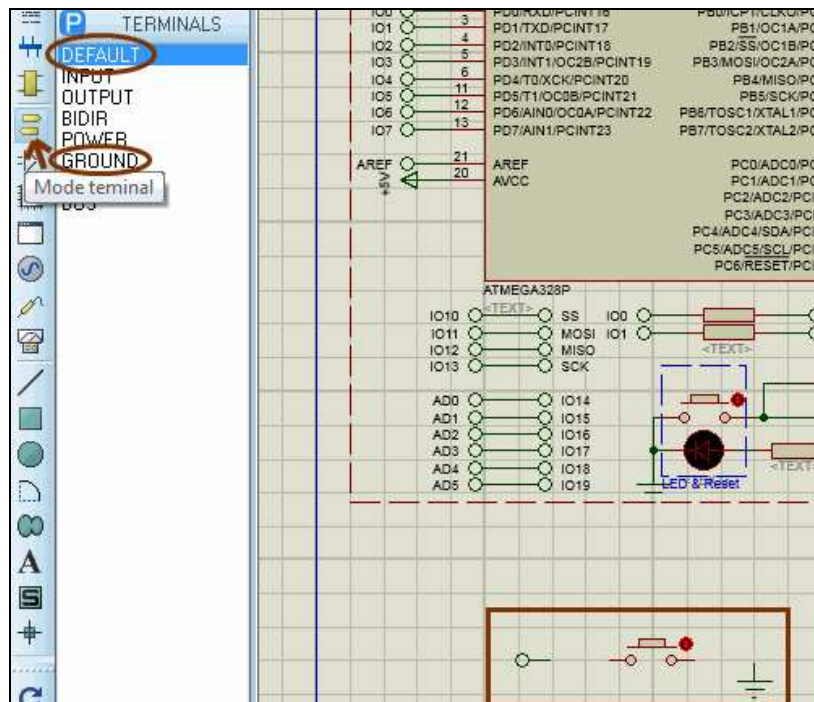
4) Avec un clic droit, vous pouvez supprimer la fenêtre active « LCD Shield – Keys ». Elle ne nous sera pas utile.



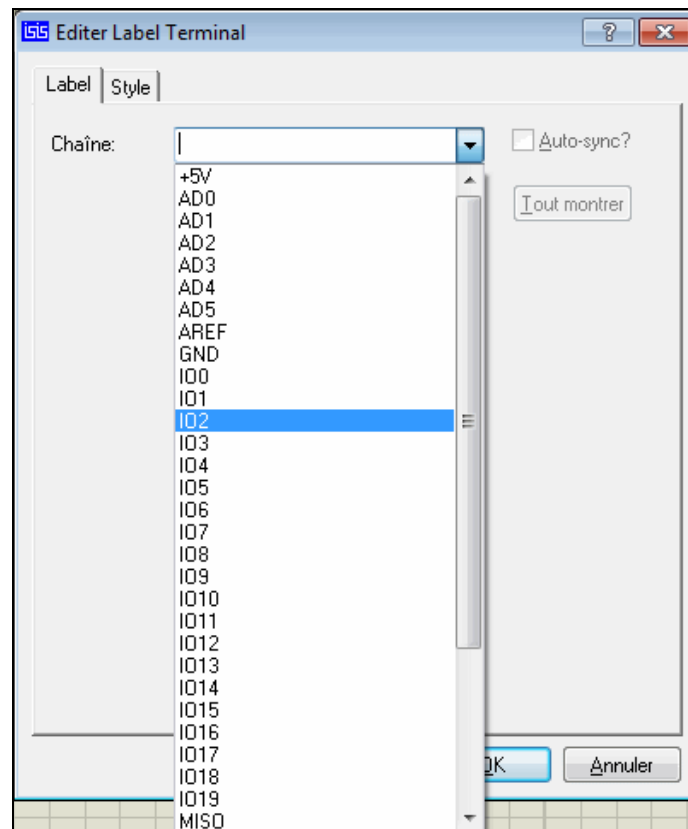
5) Cliquez sur **P** en haut à gauche de la liste des composants. Dans le champ de recherche, écrivez « button » et sélectionnez le premier élément de la liste. Cliquez sur « OK ».




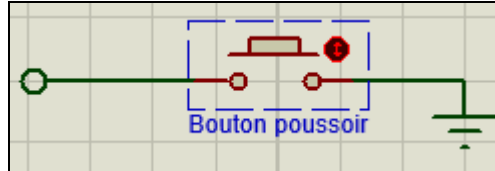
- 6) Placez un bouton dans le schéma, puis un terminal « GROUND » et un second du type « DEFAULT ». (N'oubliez pas de les câbler !)



- 7) Cliquez droit sur le label ( ) puis sur « Editer propriétés ». Dans le champ « label », sélectionnez « IO2 » qui correspond à l'entrée digitale n°2 de l'Arduino. Cliquez sur « OK ».



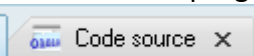
- 8) Dans la barre de gauche, cliquez sur le bouton du mode « Fenêtre active » () et dessinez un cadre autour de notre bouton poussoir. (Vous pouvez le renommer dans ses propriétés)



- 9) Notre schéma est maintenant prêt à l'emploi ! Il ne nous reste plus qu'à écrire le programme qui sera exécuté lors de la simulation.

## III. Création du programme

- 1) Allez dans la fenêtre de programmation VSM Studio en cliquant sur l'onglet en haut du

schéma : . Comme vous pouvez le remarquer, VSM a déjà préparé le squelette de votre futur programme. Nous allons à présent le remplir.

- 2) Juste au dessus de la ligne « void setup() », veuillez y inscrire ces quelques lignes :

Code à copier	Explication
// Fichiers à inclure <code>#include &lt;LiquidCrystal.h&gt;</code>	Inclut la bibliothèque propre aux écrans LCDs.
// Prototypes <code>LiquidCrystal lcd(8, 9, 4, 5, 6, 7);</code>	Définit les broches de l'Arduino sur lesquelles est branché l'écran LCD.
<code>const int BrocheBouton = 2;</code>	Constante qui indique la broche de l'Arduino connectée au bouton poussoir (IO2).
<code>int EtatBouton = 0;</code>	Variable qui stockera l'état du bouton.


- 3) À l'intérieur des accolades de l'élément « setup », veuillez y inscrire les lignes suivantes :

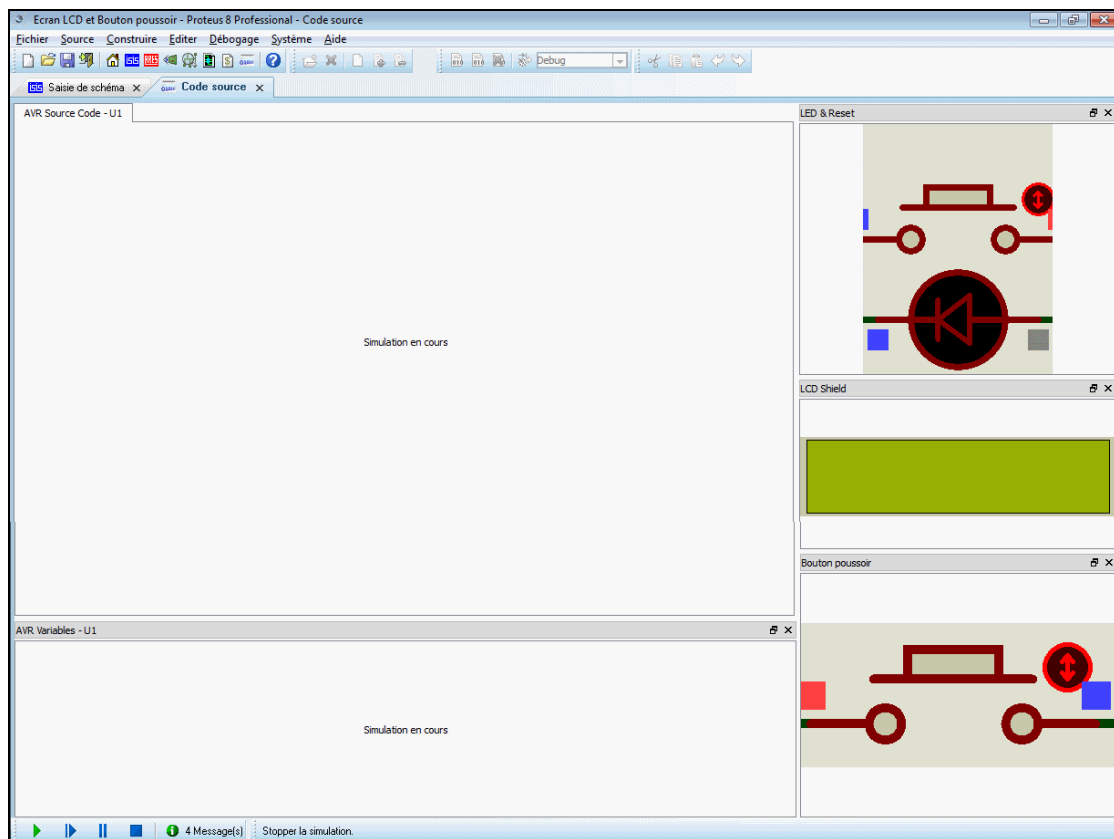
Code à copier	Explication
<code>lcd.begin(16, 2);</code>	Initialisation de l'écran LCD 16x2.
<code>lcd.print("Setup...");</code>	Message de démarrage.
<code>delay (500);</code>	Temporisation en millisecondes.
<code>lcd.clear();</code>	Effacement de l'écran.
// initialisation des broches <code>pinMode(BrocheBouton, INPUT);</code>	On indique que la broche du bouton sera une entrée.
<code>digitalWrite(BrocheBouton, HIGH);</code>	On initialise sa valeur à l'état haut.

4) À l'intérieur des accolades de l'élément « loop », veuillez inscrire ces quelques lignes :

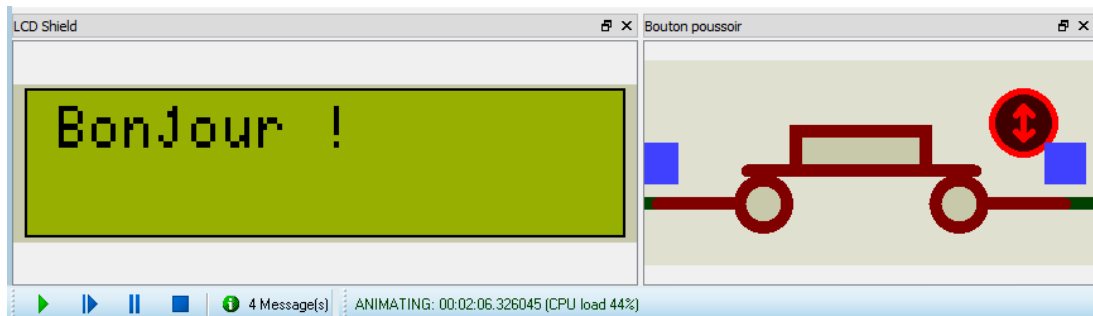
Code à copier	Explication
<code>EtatBouton = digitalRead(BrocheBouton);</code>	On stocke dans « EtatBouton » la valeur de la broche du bouton (HIGH ou LOW).
<code>if (EtatBouton == LOW) {</code>	On teste si l'on appuie sur le bouton (LOW).
<code>lcd.print("Bonjour !");</code>	Si oui, on écrit « Bonjour ! » sur l'écran.
<code>while(EtatBouton != HIGH )</code>	Tant que le bouton n'est pas relâché :
<code>EtatBouton = digitalRead(BrocheBouton);</code>	On lit la valeur de la broche du bouton en boucle.
<code>lcd.clear();</code>	Effacement de l'écran.
<code>}</code>	Fin du test « Si ».


## IV. Débogage et simulation

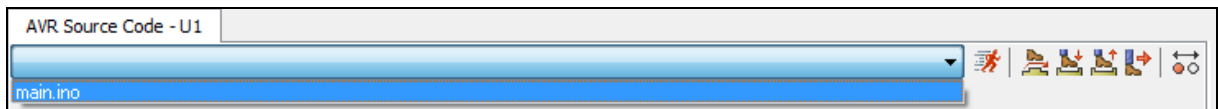
1) Une fois notre programme terminé, nous pouvons cliquer sur  pour lancer la compilation puis la simulation. Lors du premier lancement, vous verrez de nombreuses lignes défiler dans la fenêtre « Sortie VSM » en bas de l'écran. C'est normal. Proteus compile le programme que vous venez d'écrire. La simulation se lance et vous tombez sur cette page :



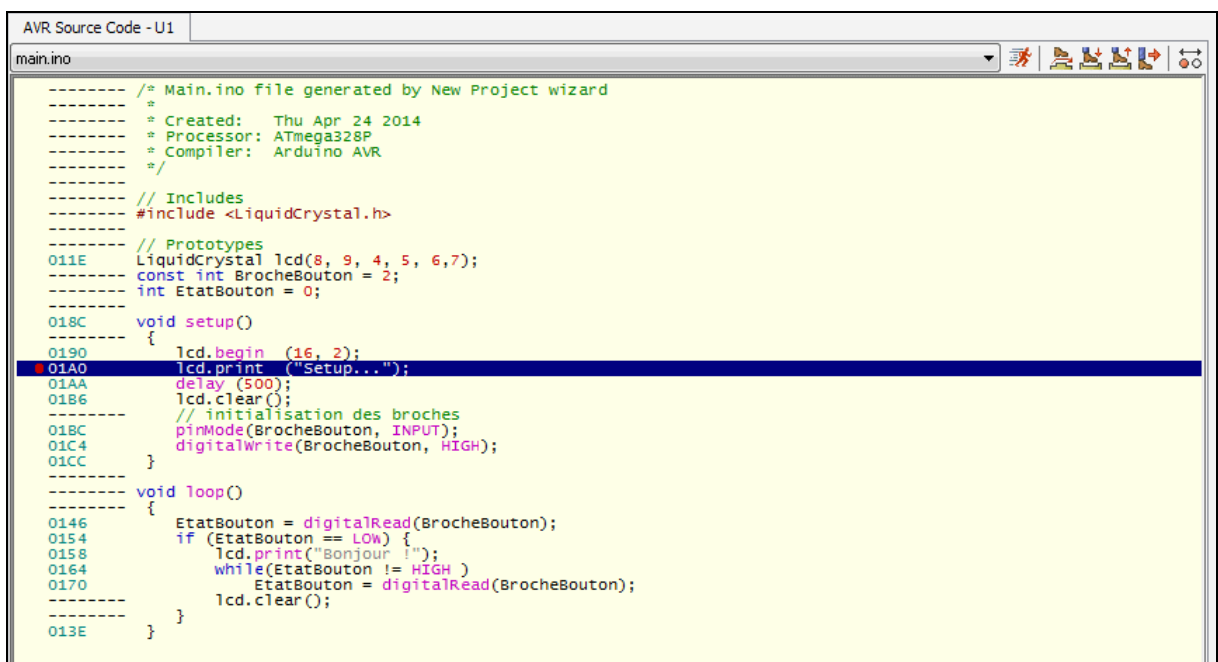
- 2) Vous remarquez que trois éléments du schéma sont représentés sur la droite. En effet, la « fenêtre active » que nous avons placée au début du TP permet de garder un œil sur les éléments importants de notre schéma pendant l'exécution de notre programme. Les deux premiers éléments sont le bouton de « Reset » de l'Arduino et l'écran LCD, ces derniers ont été ajoutés automatiquement lors de la création du projet. On retrouve en bas notre fenêtre active « Bouton poussoir » ajouté manuellement.
- 3) Vous pouvez donc appuyer sur le bouton et constater le résultat du code que vous avez écrit !





- 4) Si vous souhaitez suivre votre programme pas à pas, cliquez sur . Sélectionnez votre programme :



- 5) Placez un point d'arrêt sur la ligne de votre choix en double cliquant dessus ou à l'aide du clic droit.





- 6) Relancez la simulation et redémarrez la Carte avec le bouton de Reset.
- 7) Vous verrez le programme s'arrêter sur votre ligne avec l'indicateur . Vous pouvez avancer pas à pas avec le bouton  ou à l'aide de la touche F11.
- 8) Vous maîtrisez maintenant les bases de la simulation dans VSM d'une carte Arduino !  
À noter que le code que vous venez d'écrire peut être recopié dans le software Arduino (V1.0.5 ou plus) pour être transféré vers votre carte Uno en s'assurant de respecter le câblage qui a été choisis.

**Note :** Dans l'interface de débogage de VSM, il vous est possible de déplacer les fenêtres actives nativement positionnées sur la droite. Pour cela, sélectionnez leur entête et faites les glisser où bon vous semble. Idem pour la fenêtre des variables. Vous pouvez aussi toutes les redimensionner !