

Simulation des AVR avec SimulIDE

Introduction

Le manque d'accès à un montage physique comme une carte mère à microcontrôleur ou simplement le besoin d'accéder à un environnement de débogage justifie souvent la nécessité de se tourner vers la simulation. Pour la série des microcontrôleurs Atmel/Microchip AVR, les possibilités sont nombreuses et intéressantes sous Linux. Le type de distribution Linux utilisé influence très peu l'expérience utilisateur, car la majorité des applications de simulation sont assez stables sur toutes les variantes de Linux.

SimulIDE

Un microcontrôleur est toujours utilisé dans un système réel où des périphériques externes lui sont branchés. Il existe une solution avec une interface graphique permettant l'ajout de composants de périphériques et de plus facilement suivre le déroulement de la simulation. [SimulIDE](#) est probablement la meilleure option pour parvenir à simuler un système plus complet et de façon plus agréable. SimulIDE a une très grande collection de composants qui permet de construire des systèmes élaborés. Ce sera donc l'option privilégiée. SimulIDE possède une interface Qt qui peut être instable parfois (lire erreur de segmentation...), mais les résultats de simulation sont corrects puisque [SimAVR](#) est utilisé comme engin de simulation en arrière-plan sans qu'on s'en aperçoive.

Installation des outils de compilation pour AVR

L'installation des outils pour programmer avec les microcontrôleurs AVR est sur le site web du cours depuis longtemps. Tout de même, voici un très bref rappel de ce qu'il faut faire pour les avoir en place en une ligne si ce n'est pas déjà fait:

```
% sudo apt-get install avrdude avrdude-doc avr-libc binutils-avr  
gcc-avr gdb-avr simavr
```

La commande a été donnée ici avec les commandes pour la distribution Ubuntu. Il pourrait y avoir des ajustements à faire pour les autres distributions, mais la liste des paquets à installer restera fort probablement la même. Un gestionnaire de paquets avec interface graphique peut aussi vous venir en aide. Vous pouvez trouver le vôtre en fouillant dans le menu de démarrage de votre distribution et, en particulier, le sous-menu «administration» ce qui rend l'opération moins intimidante et tout aussi efficace.

Installation de SimulIDE

SimulIDE n'est pas agréable à installer, car on distribue la version la plus à jour dans [un format binaire compilé](#), mais voici le lien pour télécharger le paquet plus directement:

<https://www.patreon.com/file?h=29670926&i=4387728>

L'installation n'est pas intégrée à un format de paquet de distribution Linux. Le plus simple est donc d'enregistrer l'archive quelque part dans votre compte (préférentiellement à la racine) et de l'extraire par la commande suivante:

```
% tar zxvf SimulIDE_0.3.12-SR3-Lin64.tar.gz
```

Ceci devrait créer un répertoire `SimulIDE_0.3.12-SR3` qui contient lui-même un sous-répertoire `bin` contenant l'exécutable `simulide`. Par contre, il manquera des librairies Qt5 pour compléter. Donc, il faut installer ces 4 paquets supplémentaires (peut-être plus selon votre distribution Linux):

```
% sudo apt-get install libqt5script5 libqt5multimedia5  
libqt5serialport5 libqt5xml5
```

Le plus simple sera, par la suite, d'appeler directement en ligne de commande (souvent avec le chemin complet pour plus de facilité) l'exécutable `simulide` du sous-répertoire «bin». Remarquez que l'appel a été placé en arrière-plan avec le «&» en fin de ligne pour regagner l'invité (prompt) de commande bash même si l'interface graphique apparaît et que le programme continue donc de rouler:

```
% /home/<votreCompte>/SimulIDE_0.3.12-SR3/bin/simulide &
```

Il vous faudra donc adapter le chemin à la réalité de l'endroit où vous avez fait l'extraction.

Pour SimulIDE, le paquet Debian habituel est [si on consulte ce site](#) web, mais la version est plus vieille et beaucoup plus limitée. Il vaut donc mieux ne pas l'installer et ne pas s'en servir. En appelant l'exécutable par son chemin complet en ligne de commande, on est certain d'appeler le bon outil, même si une vieille version peut «traîner» ailleurs sur votre machine selon la distribution Linux utilisée...

Utilisation de SimulIDE

Sur la gauche de l'interface graphique, la bibliothèque de pièces est suffisamment garnie pour avoir un réel intérêt pour une simulation élaborée. On retrouve aussi de vrais capteurs de distance (section Micro/Sensors) comme le HC-SR04. On peut avoir sa [fiche technique](#) du fabricant SprakFun et comprendre comment il fonctionne. SimulIDE reste aussi beaucoup plus près de certaines réalités électriques. La façon de brancher les DEL et les boutons-poussoirs doit respecter les bonnes pratiques ce qui est d'un grand intérêt. On a aussi accès dans la section *Meters* à des voltmètres et des oscilloscopes.

Les programmeurs de SimulIDE ont aussi prévu des rectangles de texte et des lignes pour permettre de documenter le circuit plus facilement. Bref, on a des moyens de monter un système intéressant!

Le circuit en exemple à la figure 1 plus bas est assez proche de la réalité de la carte mère utilisée en laboratoire et des montages sur *breadboard* effectués plus tôt durant la session. Le code (souvent appelé *firmware* dans la documentation de SimulIDE) d'une simple entrée (bouton-poussoir) sur A0 et qui fait changer la sortie sur le port B a été chargé dans le

ATmega324a. On verra les «probes» changer de couleur, mais aussi une petite DEL qui est ici en série avec une petite résistance comme il se doit. Elle change aussi de couleur (si la résistance n'est pas trop élevée!) ce qui est conforme au résultat attendu. Noter aussi le bouton jaune ou rouge au haut de la fenêtre vers le centre et qui permet de démarrer ou arrêter la simulation en cours.

Le processeur choisi ici est le atmega324a qui est identique au atmega324pa sauf pour certaines réalités électriques de très bas niveau. Mais au niveau logique et de simulation numérique, les deux sont identiques.

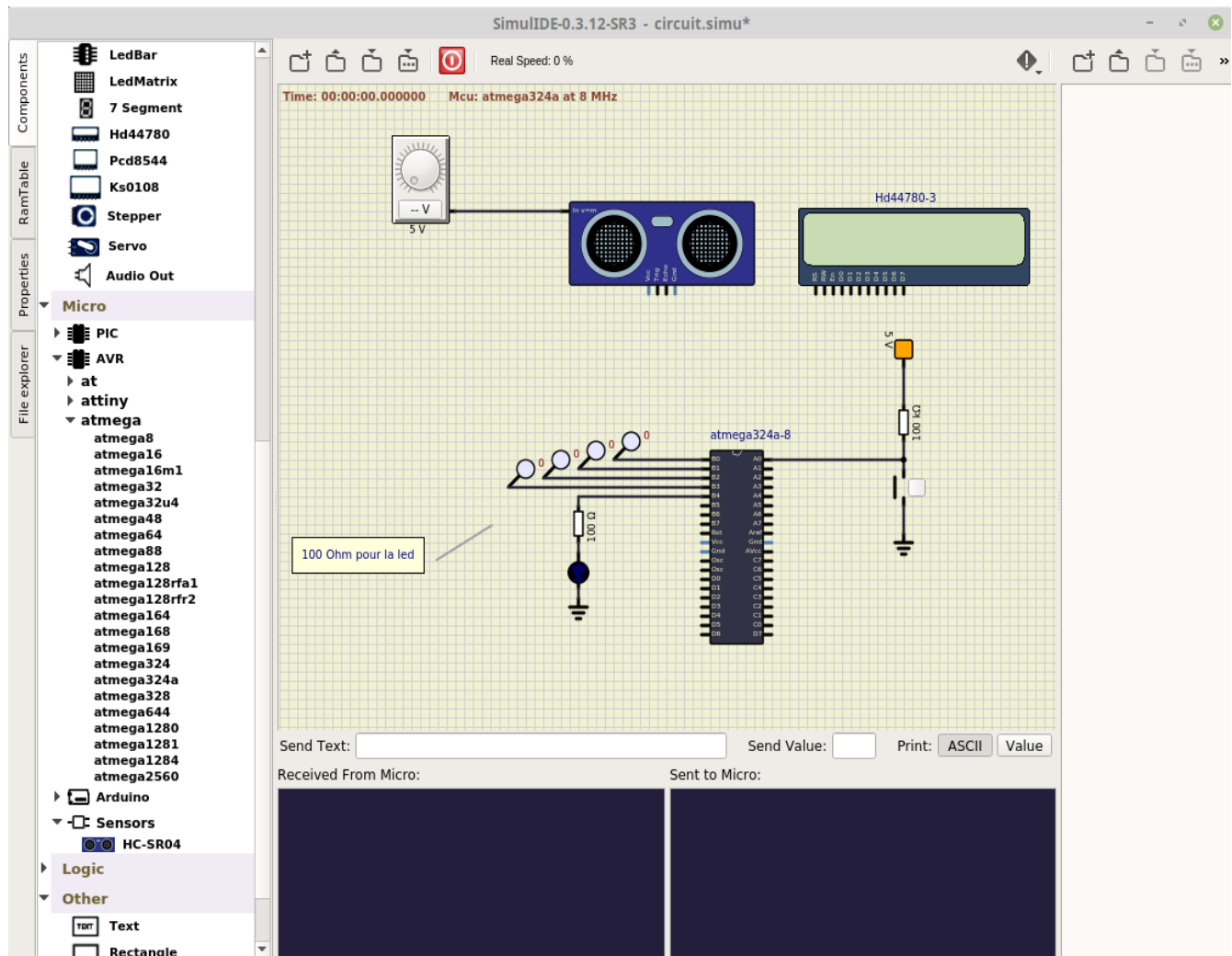


Figure 1: aperçu de SimulIDE avec un circuit simple

Bien prendre en note également qu'il faut cliquer avec le bouton droit de la souris sur le processeur AVR instantié pour charger le code à exécuter (*load firmware*) et qu'il faut prendre le fichier exécutable en format .hex généré par le Makefile modifié distribué avec ce tutoriel. Il faut aller dans «properties» du même menu pour changer la vitesse d'exécution qui est de 16 MHz par défaut.

Les composantes du haut sur le montage ne sont pas encore intégrées au circuit, mais leur fonctionnement fera plus probablement partie de travaux qui vous seront demandés par la suite.

Le circuit construit avec SimulIDE peut être sauvegardé dans un fichier dont l'extension est `.simu`. Il s'agit d'un fichier ASCII respectant le format XML qu'on peut donc éditer ou au moins regarder pour avoir une idée de ce qu'il contient. À la rigueur, un tel fichier aurait pu être également un fichier VHDL (comme en INF1500) puisqu'il s'agit d'une liste de pièces interconnectées (*netlist*). Les icônes à gauche du bouton d'arrêt/démarrage de la simulation permettent de gérer le fichier de circuit (ouvrir, sauvegarder, etc.)

Le code, quand à lui, est celui des AVR tel qu'on l'étudie depuis le début de la session, tout comme on le programmait pour le robot physique. Entre autres, la bibliothèque logicielle établie plus tôt durant la session peut toujours être utilisée sans problème.