arduino, em

Arduino : utiliser les fichiers .hex

Programmer un arduino se fait en plusieurs étapes : d'abord écrire le code, puis le compiler, c'est à dire le transformer en commandes adaptées au circuit, enfin le téléverser dans le microcontrôleur.

Le fichier compilé est du “code machine” au format hexadécimal (.hex), il n'est plus lisible pour nous autres mammifères, mais parfaitement adapté pour un circuit électronique. On ne peut pas retrouver le code d'origine depuis le fichier .hex (dans le meilleur des cas on peut le décompiler pour récupérer un programme en assembleur)

Pourquoi utiliser un fichier .hex ?

Quelques raisons d'utiliser le fichier .hex :

- le programme a été récupéré depuis un arduino, on n'en a donc pas le code source,
- pour programmer rapidement plusieurs arduino identiques avec le même programme,
- le programme n'a été diffusé que sous forme de .hex pour une quelconque raison...

Taille en mémoire

Les programmes compilés sont stockés dans la mémoire **flash** du microcontrolleur. Celui-ci est très limité comparé à ce qu'on a l'habitude d'avoir sur un PC. Pour l'Atmega328p (un microcontrolleur commun qu'on retrouve sur beaucoup d'arduinos), la mémoire flash disponible est de 32Ko. Il faudra donc faire attention à la taille de nos programmes.

Lors de la compilation du code source dans l'IDE Arduino, la taille du programme compilé est donnée dans la fenêtre de débogage. Si toutefois vous utilisez un autre environnement pour programmer votre microcontrolleur, vous pouvez utiliser la commande linux “*size*” pour connaître la taille que prendra le fichier .hex en mémoire flash.

```
$ size vamp1.hex
  text      data       bss       dec       hex    filename
     0     29766         0     29766      7446    vamp1.hex
```

Notez que cette taille (29Ko dans notre exemple) est différente de la taille du fichier .hex sur notre disque dur, comme indiqué par la commande suivante :

```
$ du -h vamp1.hex
 84K    vamp1.hex
```

Comment téléverser un fichier .hex ?

Il faut retrouver le chemin d'avrdude, le logiciel qui envoie le code machine dans le microcontrôleur

- dans les préférences, activer “Afficher les résultats détaillés pendant le téléversement”
- compiler et téléverser un programme
- noter la commande complète avrdude

Exemple :

```
/home/emoc/arduino-1.8.5/hardware/tools/avr/bin/avrdude -C/home/emoc/arduino-1.8.5/hardware/tools/avr/etc/avrdude.conf -v -patmega328p -
carduino -P/dev/ttyUSB0 -b57600 -D -Uflash:w:/tmp/arduino_build_700970/Test.ino.hex:i
```

Dans ce cas précis, il s'agit d'un code compilé pour arduino nano (ATmega328P)

On peut découper cette ligne pour l'expliquer :

/home/emoc/arduino-1.8.5/hardware/tools/avr/bin/avrdude		chemin vers le logiciel avrdude

-C	/home/emoc/arduino-1.8.5/hardware/tools/avr/etc/avrdude.conf	chemin vers le fichier de configuration
-v		exécution de la commande “bavarde” (*verbose*)
-p	atmega328p	modèle du microcontrôleur à programmer
-c	arduino	programmeur associé, ici : Atmel AppNote AVR109 Boot Loader
-P	/dev/ttyUSB0	port USB série
-b	57600	vitesse de transfert (*baudrate*)
-D		désactiver l'effacement automatique de la mémoire flash
-U	flash:w:/tmp/arduino_build_700970/Test.ino.hex:i	voir ci-dessous

Pour la dernière option (-U) la commande signifie : utiliser la mémoire flash (flash), pour écrire (w), un fichier .hex, encodé en hexadécimal intel (i)

Avec cette commande il est donc possible de flasher n'importe quel fichier hex compilé pour le même microcontrôleur, sans passer par l'IDE arduino. Il suffit de respecter les mêmes options et de faire pointer vers le fichier hex approprié.

Exemple :

```
/home/emoc/arduino-1.8.5/hardware/tools/avr/bin/avrdude -C/home/emoc/arduino-1.8.5/hardware/tools/avr/etc/avrdude.conf -v -patmega328p -carduino -P/dev/ttyUSB0 -b57600 -D -Uflash:w:/home/emoc/test/backup.hex:i
```

Créer un fichier .hex

Dans le cas ou l'on peut accéder au code source, il est possible de créer le fichier .hex pour le conserver depuis le menu “Croquis/exporter les binaires compilées”, le fichier .hex sera alors enregistré dans le dossier du sketch sous deux formes : avec ou sans bootloader. Il faudra choisir la bonne version selon les situations : en général, s'il s'agit d'une carte arduino la versio sans bootloader est adaptée.

Récupérer un fichier hex

On peut télécharger le code machine intégré dans un arduino programmé en utilisant avrdude, exemple ci-dessous pour récupérer le code machine d'un arduino nano

```
/home/emoc/arduino-1.8.5/hardware/tools/avr/bin/avrdude -C/home/emoc/arduino-1.8.5/hardware/tools/avr/etc/avrdude.conf -v -patmega328p -carduino -P/dev/ttyUSB0 -b57600 -D -Uflash:r:/home/emoc/test/backup.ino.hex:i
```

La différence principale étant le r dans la dernière option (r pour read!)

Cela ne fonctionne pas avec tous les modèles d'arduino comme ceux basés sur l'ATmega32U4, c'est lié à des différences de bootloader...

Sources

Par ici :

- https://www.nongnu.org/avrdude/user-manual/avrdude_4.html#Option-Descriptions
- https://arduino.stackexchange.com/q/48431
- https://forum.arduino.cc/index.php?topic=403201.msg2786267#msg2786267
- récupération de hex et problème d'ATmega32U4 : https://arduino.stackexchange.com/a/21534

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/ressource/electronique/arduino/utiliser_fichier_hex**
Article mis à jour: **2021/07/14 10:41**