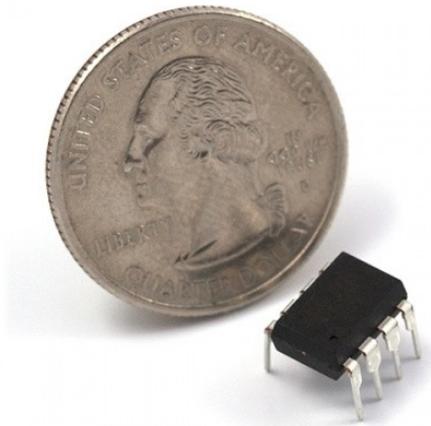


Guide d'utilisation du programmeur AVR

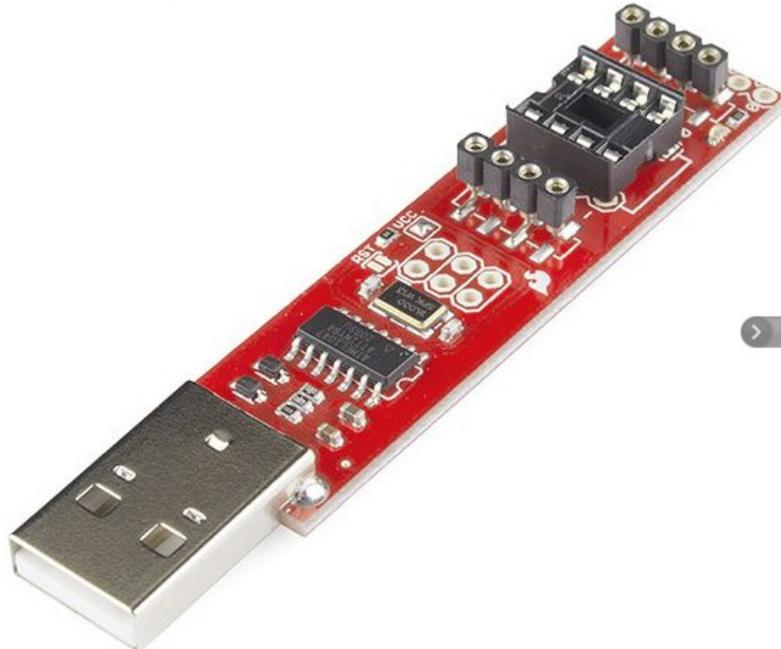
Introduction

Arduino est génial. Les cartes sont solides, le langage de programmation et l'IDE sont faciles, et la communauté est géniale. Mais pour beaucoup de projets électroniques, un Arduino est excessif. Si vous vous contentez de faire clignoter quelques LEDs, et de lire un seul capteur, vous pouvez faire le travail plus petit et moins cher en utilisant un simple circuit intégré, comme l'ATtiny85.



Notre héros ! L'ATtiny85

Malheureusement, l'ATtiny85 n'a pas de plate-forme de développement connue et omniprésente comme l'Uno ou le Leonardo d'Arduino. Et 8kB d'espace de programmation ne laisse pas beaucoup de place pour un bootloader, donc un programmeur supplémentaire est généralement nécessaire. De plus, l'Arduino standard ne prend pas en charge cette puce. Cela ne veut pas dire que la programmation de l'ATtiny85 dans Arduino n'est pas possible !



Tiny AVR Programmer

Le Tiny AVR Programmer est un programmeur AVR général, mais il est spécifiquement conçu pour permettre une programmation rapide et facile des ATtiny85 (ainsi que des 45) par rapport au programmeur AVR de poche. Il dispose d'une prise intégrée, où le petit circuit intégré à 8 broches peut être branché et directement programmé. Pas besoin de fils ou de soudure ! Une fois l'ATtiny85 programmé, il suffit de le retirer du programmeur et de le coller sur une planche à pain ou une carte de prototypage.

Le Tiny AVR Programmer peut également être utilisé comme un programmeur AVR à usage général. Il peut programmer directement presque tous les AVR (y compris l'ATmega328 et l'ATmega32U4) qu'ils soient sur des cartes Arduino ou sur une breadboard.

Sujets abordés dans ce tutoriel

Dans ce guide de branchement, nous allons montrer comment vous pouvez programmer les ATtiny85 en utilisant le Tiny AVR Programmer et Arduino. Nous couvrirons tous les aspects, de l'installation du pilote aux conseils de programmation Arduino.

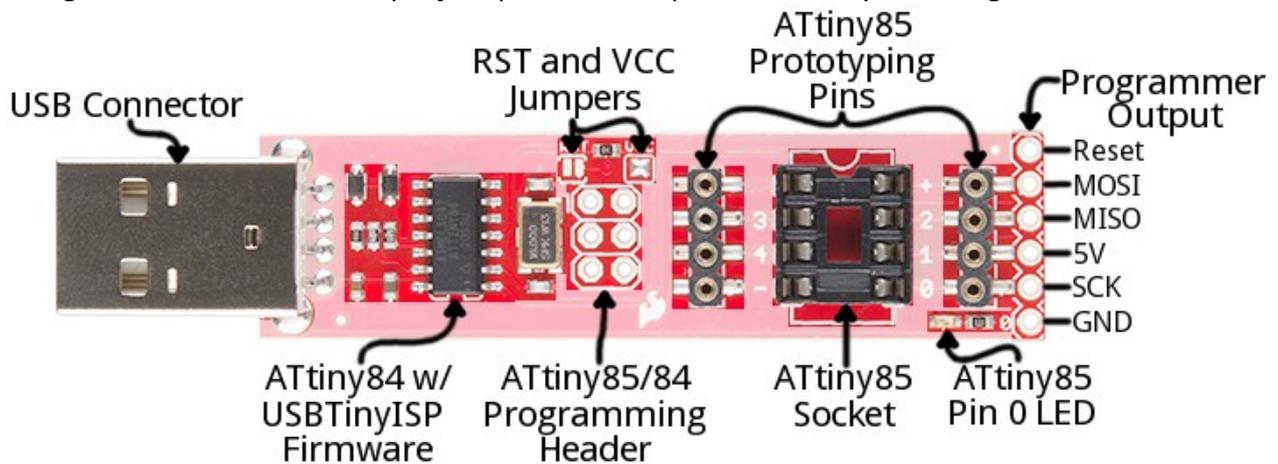
Matériel nécessaire

En plus du Tiny AVR Programmer, vous aurez besoin des éléments suivants pour suivre ce tutoriel :

- ATtiny85 -- Pour être programmé par le programmeur. N'oubliez pas que vous pouvez également l'utiliser pour flasher d'autres puces AVR comme l'ATtiny84 !
- Un ordinateur ou un portable avec :
 - Un port USB libre. Un hub USB devrait également fonctionner.
 - L'IDE Arduino est installé.
- Optionnel :
 - Câble d'extension USB -- Si votre port USB est hors de portée, ceci peut aider à rendre le programmeur plus facile à atteindre.
 - Clip de test IC - SOIC 8-Pin -- Si vous utilisez un ATtiny monté en surface, ce petit clip pratique facilite la programmation du microcontrôleur !
 - Fils de pontage Premium M/F -- Utiles si vous connectez le Tiny AVR Programmer à un autre microcontrôleur AVR qui n'est pas l'ATtiny85 ou le IC Test Clip.

Vue d'ensemble de la carte

L'image ci-dessous donne un aperçu rapide des composants du Tiny AVR Programmer :

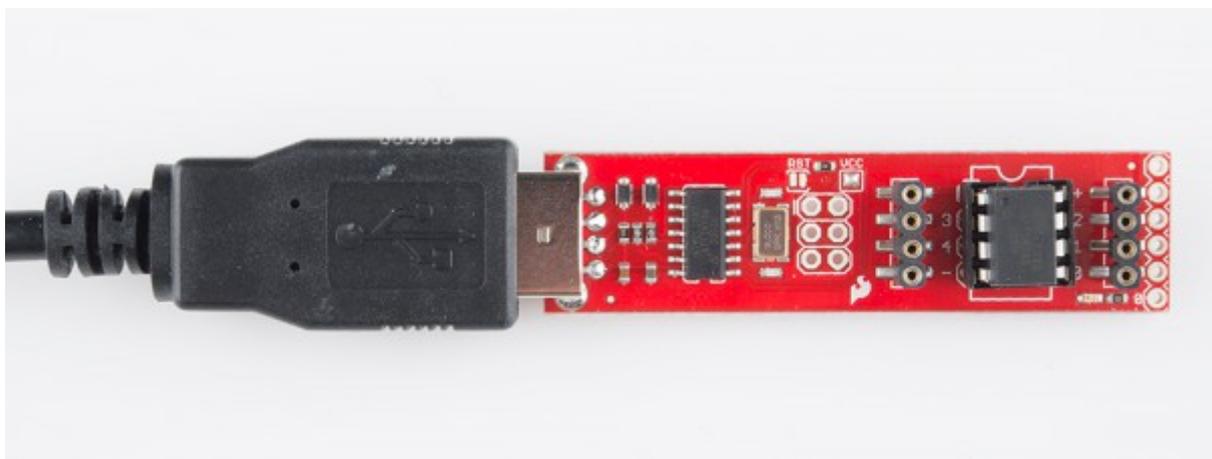


Le "cœur" du Tiny AVR Programmer est un ATtiny84 (à ne pas confondre avec le 85), -- la puce de montage en surface à 16 broches -- qui est livré préprogrammé avec un micrologiciel qui le fait ressembler à un programmeur d'AVR. A moins que vous n'écriviez un firmware ISP AVR personnalisé, vous ne devriez jamais avoir à manipuler cette puce. C'est une boîte noire. Les données du programme lui parviennent de votre ordinateur, via USB, et il restitue la bonne séquence d'octets pour charger ce programme dans votre ATtiny85.

Dans ce tutoriel, nous nous intéresserons principalement aux composants situés sur la moitié droite de la carte. La prise de programmation de l'ATtiny85, la LED de la broche 0, et les broches de prototypage.

Support ATtiny85 et broches de prototypage

Le socle et les broches réparties sur les côtés sont ce qui rend le Tiny AVR Programmer unique. La prise à 8 broches convient à la fois aux boîtiers DIP ATtiny85 et ATtiny45. Il suffit de brancher votre circuit intégré à programmer dans cette prise, et vous pouvez commencer à programmer !

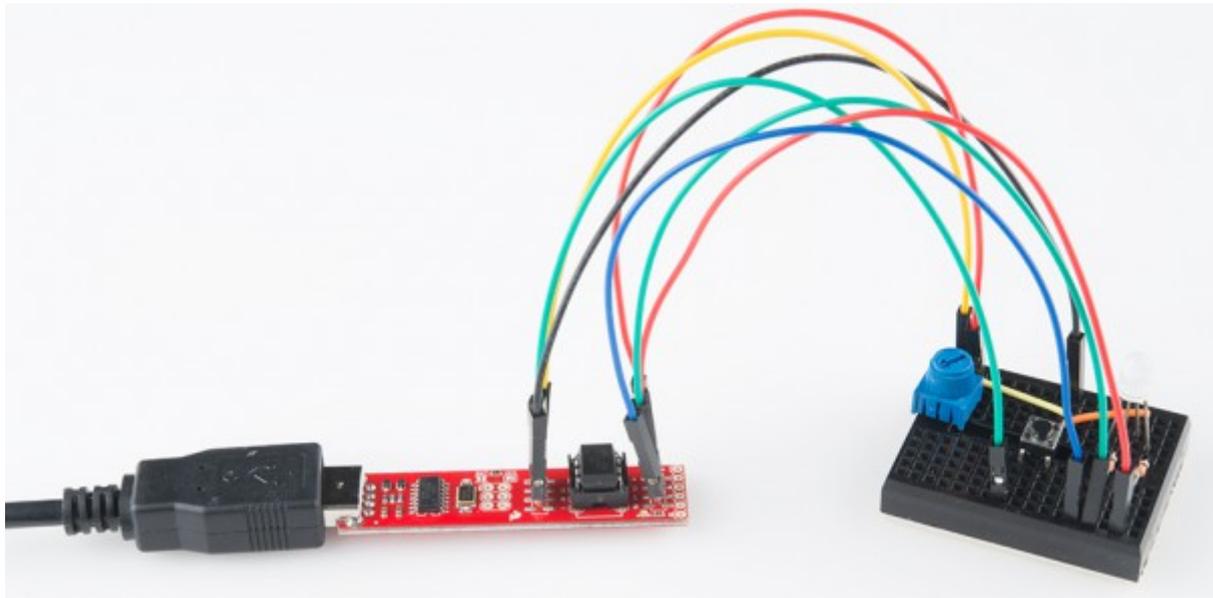


Un petit programmeur AVR avec un ATtiny85 inséré

Lorsque vous branchez votre ATtiny dans son support, prenez note de l'encoche sur la prise et de la sérigraphie blanche sur le circuit imprimé. Cela doit correspondre à la polarité de l'ATtiny85. Habituellement, l'ATtiny85 a un point à côté de la broche 1 du circuit intégré, ce point doit être placé vers l'encoche.

Les indications +, - et numériques sur le côté de la prise font référence aux numéros de broches et aux entrées d'alimentation en tension de l'ATtiny85. Ces numéros de broches peuvent être appelés dans l'IDE Arduino comme nous le montrerons plus loin dans ce tutoriel.

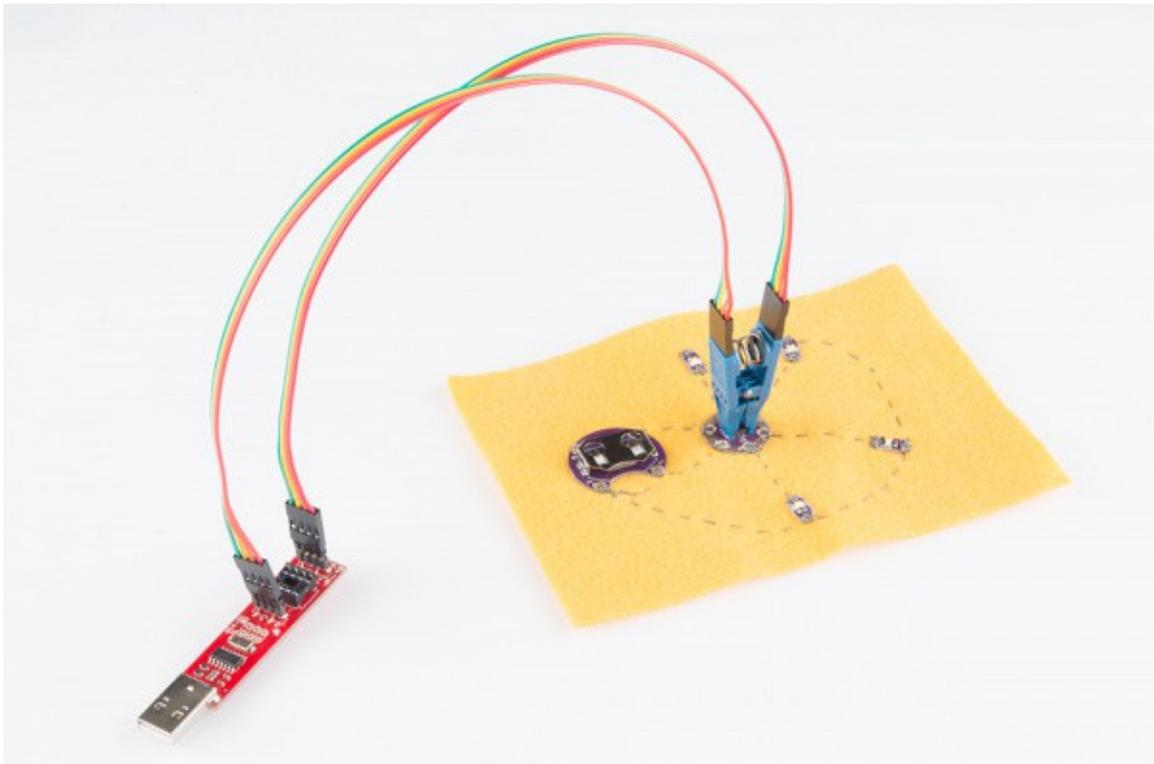
Les connecteurs à 4 broches de chaque côté de la prise permettent de prototyper la sortie de l'ATtiny85 vers des circuits externes. Vous pouvez facilement brancher des fils de liaison mâles sur ces broches, qui peuvent être acheminés vers des cartes à de prototypage ou d'autres dispositifs.



Prototype d'un ATtiny85 relié à un potentiomètre (entrée analogique), un bouton (entrée numérique) et une LED RVB (sortie analogique/numérique)

Note : Les fonctionnalités I2C et SPI de l'ATtiny85 ne peuvent pas être utilisées dans cette méthode parce que les broches sont toujours liées aux broches SPI de l'ATtiny84 utilisé pour programmer la puce. Le circuit intégré doit d'abord être retiré du programmeur.

Les connecteurs à 4 broches peuvent également être utilisés pour connecter des ATtiny85 (CMS) montés en surface ou d'autres microcontrôleurs AVR sur des breadboards.



Tiny AVR Programmer connecté à un ATtiny85 monté en surface et cousu sur du tissu pour reprogrammer un LilyTiny

Enfin, il y a une LED ambre intégrée connectée à la broche 0 de l'ATtiny85. Ceci est très utile lorsque vous téléchargez le sketch de clignotement "Hello, world" sur un ATtiny85.

Cela couvre les aspects fondamentaux du Tiny AVR Programmer. Si vous avez l'intention de faire des choses plus avancées avec la carte, ou si vous voulez simplement en savoir plus, n'hésitez pas à lire la suite.

Broches de programmation de sortie

Le Tiny ISP Programmer n'est pas limité aux ATtiny85. C'est un programmeur AVR à part entière. Cette rangée de six broches peut être connectée à d'autres AVR via les connecteurs ISP standards 2x3 ou 2x5 broches. Vous pouvez, par exemple, connecter ces broches à votre Arduino Uno, Leonardo, etc. pour re-flasher un bootloader, ou télécharger du code à l'aide d'un programmeur.

Reportez-vous aux libellés des broches dans l'image ci-dessus si vous connectez le Tiny AVR Programmer à une autre puce AVR. La plupart des cartes de développement AVR disposent d'un en-tête de programmation 2x3 ou 2x5, dont les brochages sont les suivants :

Il suffit de faire correspondre les libellés du Tiny Programmer aux broches de votre carte/puce AVR, et de vous préparer à programmer !



Les cavaliers

Il y a deux cavaliers sur la face supérieure du Tiny AVR Programmer : l'un est étiqueté RST et l'autre est VCC. Ces deux cavaliers affectent le connecteur ICSP (in-circuit system programmer) 2x3 non rempli au milieu de la carte. A moins que vous n'envisagiez de reprogrammer l'ATtiny84 intégré, ces cavaliers et ces broches peuvent généralement être ignorés.

Le cavalier VCC est normalement connecté. Il contrôle le flux d'alimentation vers la broche VCC sur l'en-tête ICSP. Lorsqu'il est fermé, l'alimentation provenant de l'USB circulera vers l'en-tête ICSP. Lorsqu'il est ouvert, vous devrez fournir une alimentation externe à cette broche.

Le cavalier RST est normalement ouvert. Lorsqu'il est fermé, ce cavalier connecte la broche de réinitialisation de l'ATtiny84 à l'en-tête de programmation 2x3. Si vous avez besoin de reprogrammer l'ATtiny84 (ce qui, dans les cas d'utilisation standard, ne devrait pas être le cas), vous devrez fermer ce cavalier pour pouvoir le programmer.

Assez parlé. Commençons à utiliser le programmeur. Dans les pages suivantes, nous allons couvrir l'installation du pilote (pour les utilisateurs de Windows) et montrer comment vous pouvez utiliser le Tiny AVR Programmer pour programmer un ATtiny85 dans Arduino.

Installation du pilote

Avant de commencer à utiliser le Tiny AVR Programmer, vous devrez peut-être l'installer sur votre ordinateur. Si vous utilisez une machine Mac ou Linux, vous n'avez pas besoin d'installer de pilotes. Branchez simplement la carte et passez à la page Programmation dans Arduino.

Si vous utilisez une version quelconque de Windows, vous devez suivre quelques étapes avant de pouvoir rejoindre vos camarades Mac/Linux. Cette page contient deux séries d'instructions pour l'installation des pilotes. La première est la méthode la plus simple et la plus rapide, et devrait convenir à la plupart des utilisateurs. La seconde procédure d'installation n'est nécessaire que si la première échoue : elle adopte une approche plus manuelle de l'installation des pilotes.

Installer automatiquement les pilotes avec Zadig

Pour commencer, branchez le Tiny AVR Programmer sur votre ordinateur. Lors de la connexion initiale de la carte, Windows essaiera d'installer automatiquement les pilotes. Certains ordinateurs auront de la chance, mais la plupart afficheront un message vous informant que l'installation du pilote a échoué.

Cliquez sur le lien ci-dessous pour télécharger le logiciel et les pilotes Zadig :

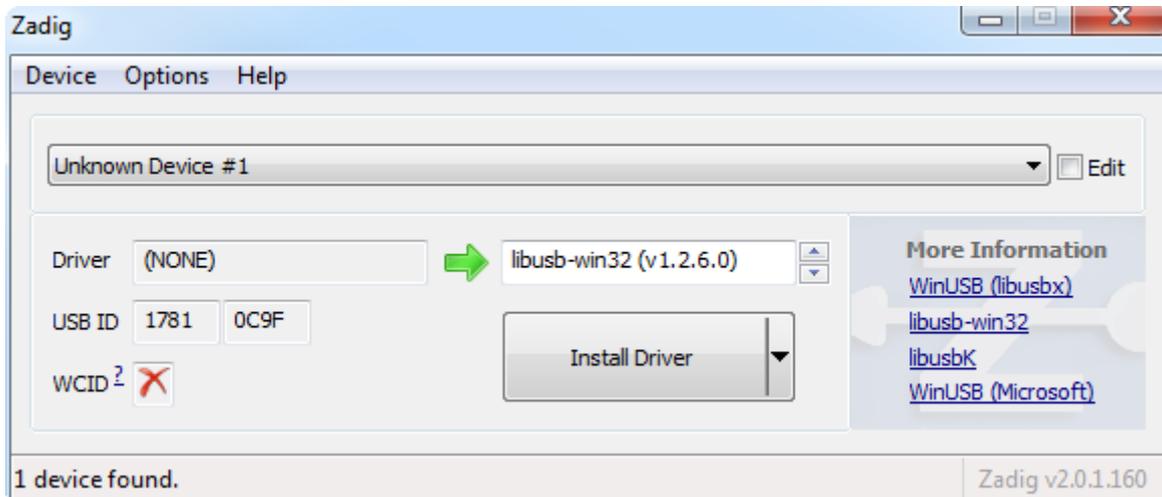
[Télécharger les drivers ZADIG USBTINY \(ZIP\)](#)

Utilisez votre décompresseur préféré pour extraire le fichier ZIP. N'oubliez pas l'emplacement du dossier extrait !

Une fois que vous avez branché le Tiny AVR Programmer sur votre ordinateur et que votre machine a exécuté le processus de vérification et d'échec de l'installation des pilotes, allez dans le dossier

"zadig_v2.0.1.160" que vous venez de décompresser. Exécutez ensuite le logiciel zadig.exe.

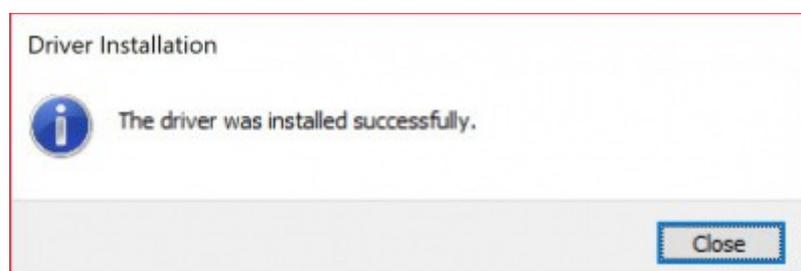
Zadig est un outil merveilleux qui peut installer les pilotes sur pratiquement toutes les plateformes Windows existantes. À l'ouverture du programme, vous devriez être accueilli par une fenêtre comme celle-ci :



Il y a quelques options à vérifier avant d'installer le pilote :

- Sélectionner le périphérique -- La boîte de dépôt supérieure contrôle le périphérique pour lequel vous souhaitez installer le pilote. Si tout va bien, vous n'avez qu'une seule option ici, quelque chose comme "Unknown Device #1". Si vous avez plus d'une option, vérifiez votre gestionnaire de périphériques pour voir si vous pouvez déterminer lequel est le bon (brancher et débrancher un périphérique aide généralement).
- Sélectionnez le pilote -- Cliquez sur les flèches de cette boîte jusqu'à ce que vous tombiez sur libusb-win32 (vx.x.x.x), c'est le pilote que nous voulons installer.

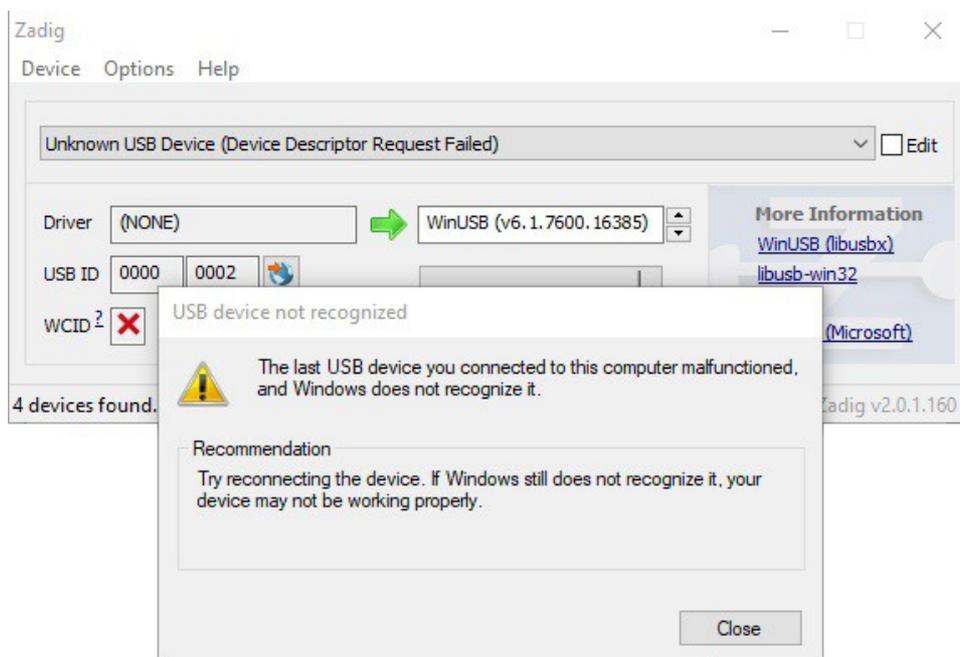
Après avoir vérifié ces deux sélections, cliquez sur " Install Driver " (Installer le pilote). Le processus d'installation peut prendre quelques minutes, mais après avoir regardé la barre de défilement défiler un nombre incalculable de fois, vous devriez être accueilli par un message "The driver was installed successfully" (Le pilote a été installé avec succès).



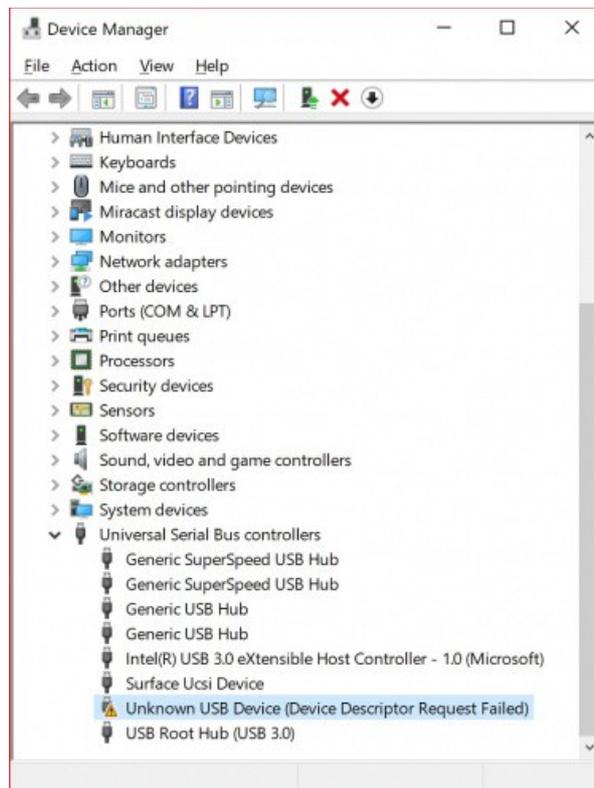
Problèmes d'installation du pilote Zadig

Après avoir installé les pilotes, votre ordinateur peut réagir en indiquant que le périphérique n'a pas été installé correctement. Voici deux méthodes pour résoudre les problèmes liés aux pilotes lors de l'installation avec Zadig.

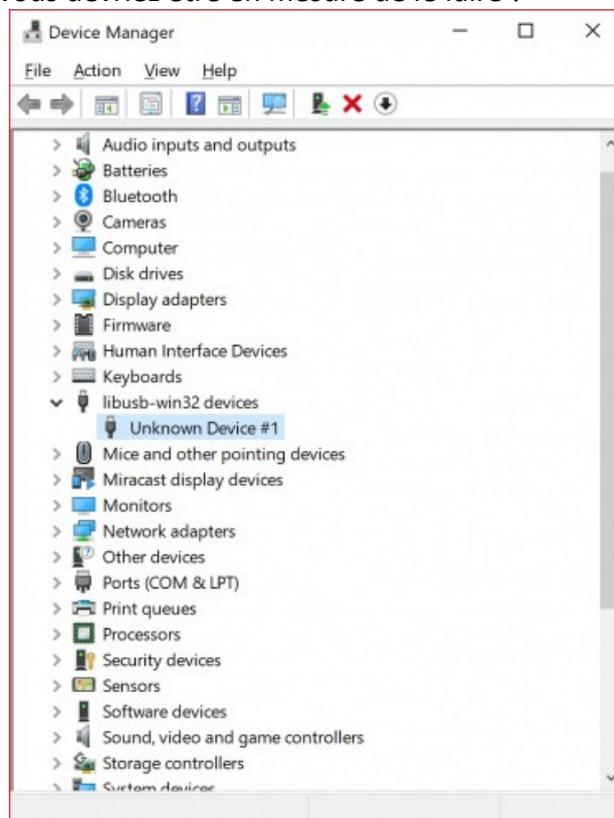
□ **Conseil de dépannage** : Dans ce cas, les pilotes WinUSB ont été sélectionnés au lieu des pilotes libusb-win32. Pour remédier à ce problème, il suffit de suivre à nouveau le guide pour réinstaller les bons pilotes libusb-win32.



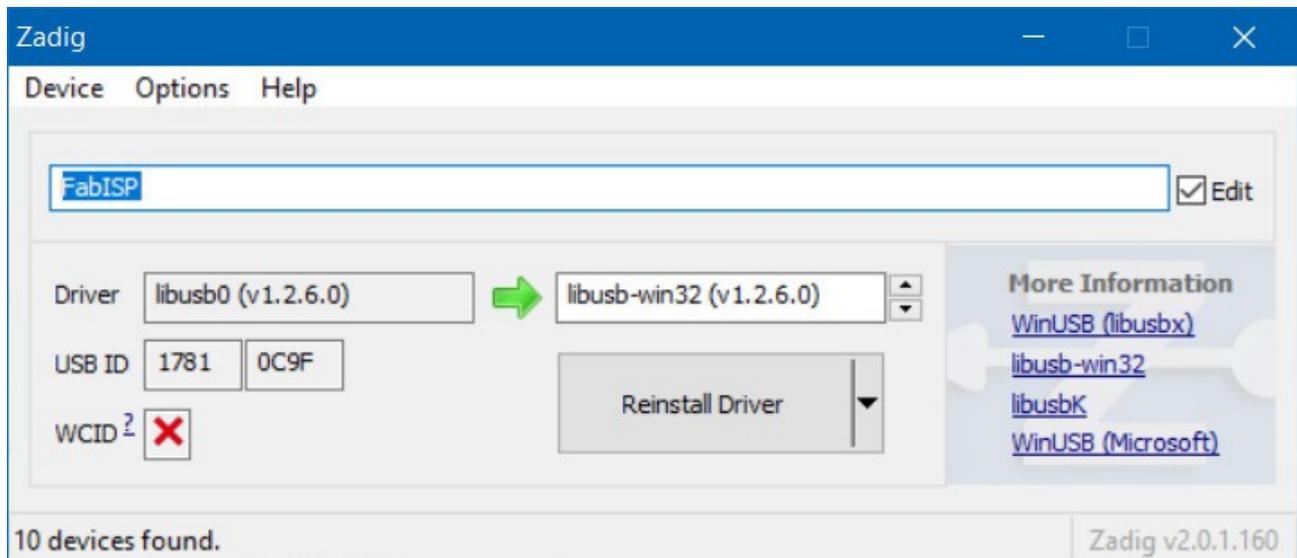
□ **Conseil de dépannage** : Dans d'autres cas, il peut également s'initialiser quelque part dans votre gestionnaire de périphériques en tant que périphérique USB inconnu (Device Descriptor Request Failed), même si vous avez installé les pilotes corrects :



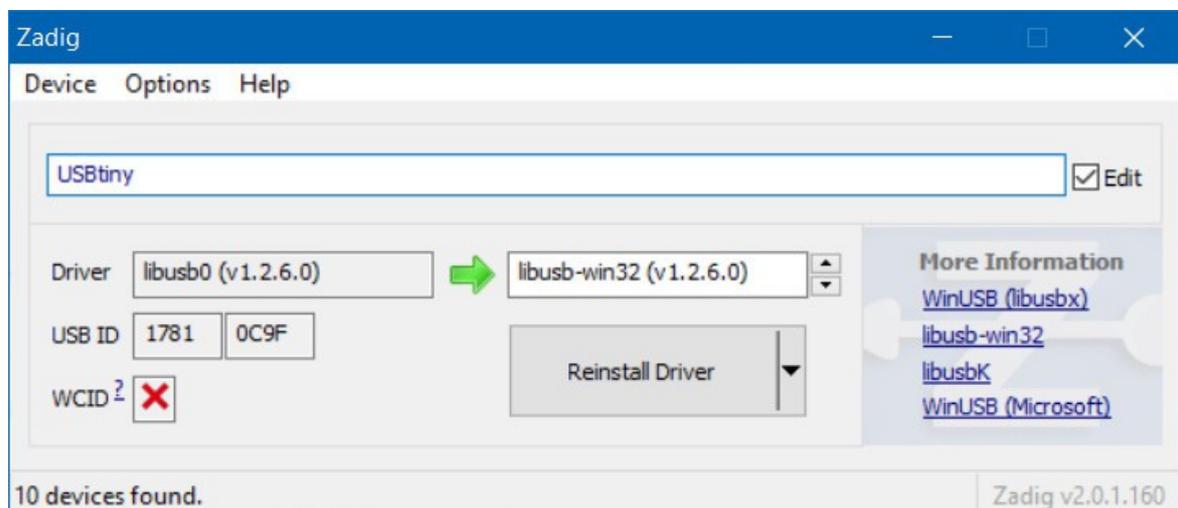
Essayez de débrancher et de rebrancher le Tiny AVR Programmer dans votre port USB. Ou remplacez votre câble d'extension USB par un câble de qualité connue. Dans certains cas, votre Tiny AVR Programmer peut apparaître sous les périphériques libusb-win32 en tant que Unknown Device #1. Si c'est le cas, vous devriez être en mesure de le faire !



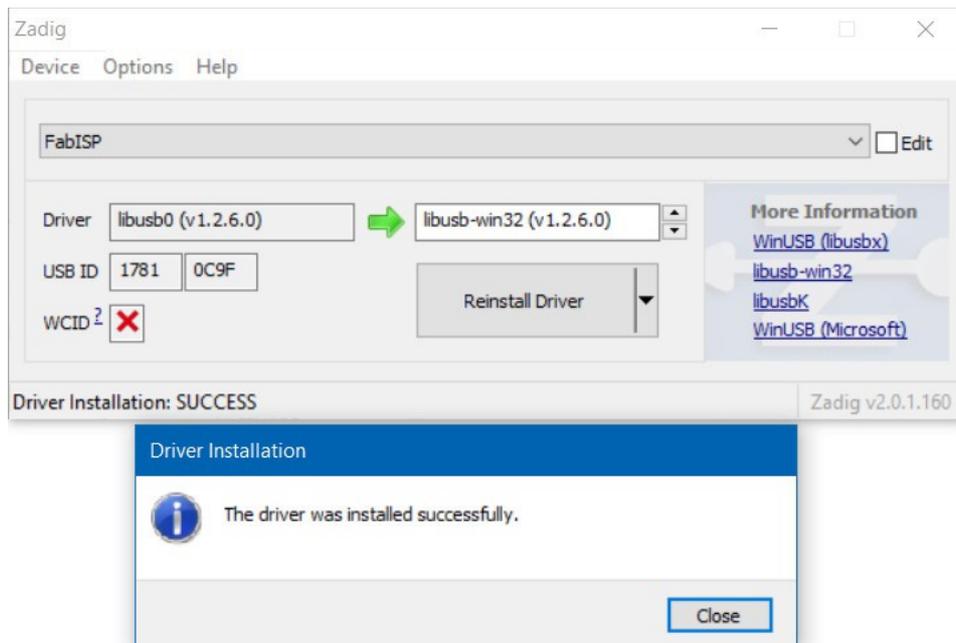
Bravo ! Vous avez réussi à installer les pilotes sur votre ordinateur. Cependant, le pilote s'affiche toujours comme un Périphérique Inconnu #1 (dans certains cas comme l'image ci-dessous, le Tiny AVR Programmer peut s'afficher comme des périphériques libusb-win32 > FabISP). Mais vous savez ce que c'est ! Vous pouvez utiliser le logiciel Zadig pour renommer le port USB si vous le souhaitez. Lorsque votre programmeur est connecté à votre ordinateur et que le logiciel est ouvert, naviguez jusqu'au port du programmeur. Cochez la case située à côté de Modifier.



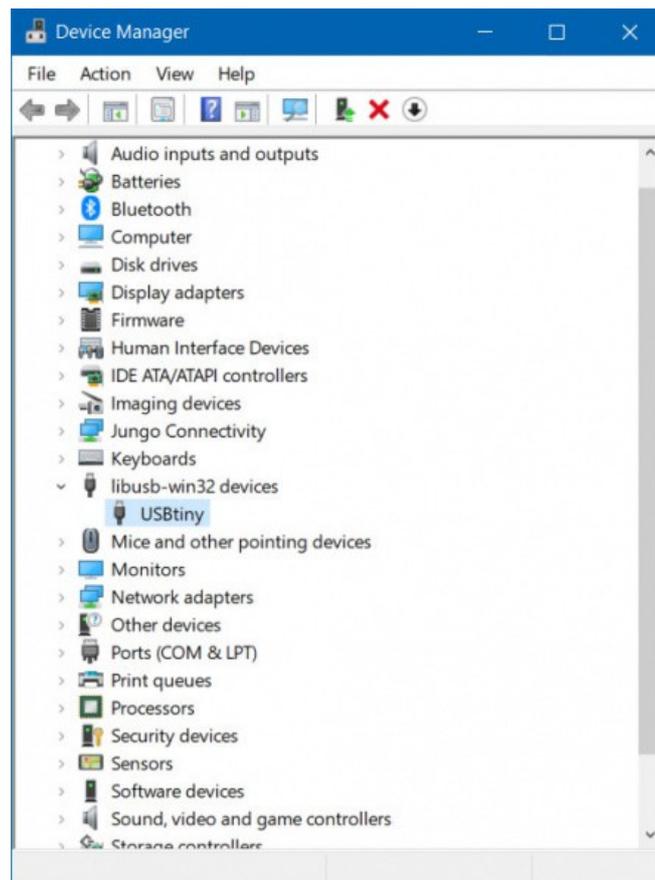
Tapez le nom de votre port. Il peut s'agir de "USBtiny" ou dans ce cas. Assurez-vous que le bon pilote est sélectionné.



Cliquez sur Réinstaller le pilote. Le pilote sera réinstallé et vous devriez voir le même message indiquant que les pilotes ont été installés avec succès. Il se peut que vous deviez débrancher et rebrancher le programmeur à votre ordinateur pour lui donner le temps de se réactualiser.



Ouvrez votre gestionnaire de périphériques et vous devriez voir le périphérique renommé !



Si vous avez réussi, fermez le programme Zadig et passez à la section suivante !

GUIDE DE PROGRAMMATION

Si Zadig n'a pas fonctionné pour vous, consultez les instructions [suivantes](#) pour vous aider à installer manuellement les pilotes.

Respirez tranquillement maintenant ! Une fois que vous avez installé les pilotes USBtiny sur votre ordinateur, vous ne devriez plus jamais avoir à le faire. Il est maintenant temps de programmer quelque chose !

Programmation dans Arduino

Tout le monde aime Arduino ! Le langage simplifié rend la programmation des AVR et des microcontrôleurs plus sophistiqués incroyablement facile. Malheureusement, Arduino n'a pas de fonctionnalité intégrée pour programmer les petits AVR, mais cela ne veut pas dire que nous ne pouvons pas l'ajouter !

Sur cette page, nous allons passer en revue toutes les étapes nécessaires pour activer la programmation de l'ATtiny45/85 dans Arduino, en utilisant le Tiny AVR Programmer.

Installation du module complémentaire Attiny

L'étape suivante consiste à installer le module complémentaire Attiny. Les étapes suivantes (1a et 1b) expliquent comment installer manuellement les fichiers de la carte ATtiny pour Arduino.

☐ **Astuce** : Pour les débutants, vous pouvez procéder à une installation automatique en utilisant le gestionnaire de cartes Arduino en suivant les instructions dans "[Installer le support ATtiny dans Arduino v1.6.4+](#)".

Télécharger l'addon Attiny

Pour ajouter manuellement les ATtiny au menu standard de la carte de l'IDE Arduino, vous devez ajouter quelques fichiers qui aident à définir le matériel. Les dernières définitions du matériel ATtiny sont conservées dans un [dépôt sur GitHub](#).

Vous pouvez les télécharger à partir de là, ou simplement cliquer sur les liens archivés ci-dessous (note : Les fichiers sont différents selon la version d'Arduino que vous utilisez) :

- [ATtiny for Arduino 1.0.x](#)
- [ATtiny for Arduino 1.6.x](#)

Extrayez le dossier ZIP, et n'oubliez pas où vous l'avez mis !

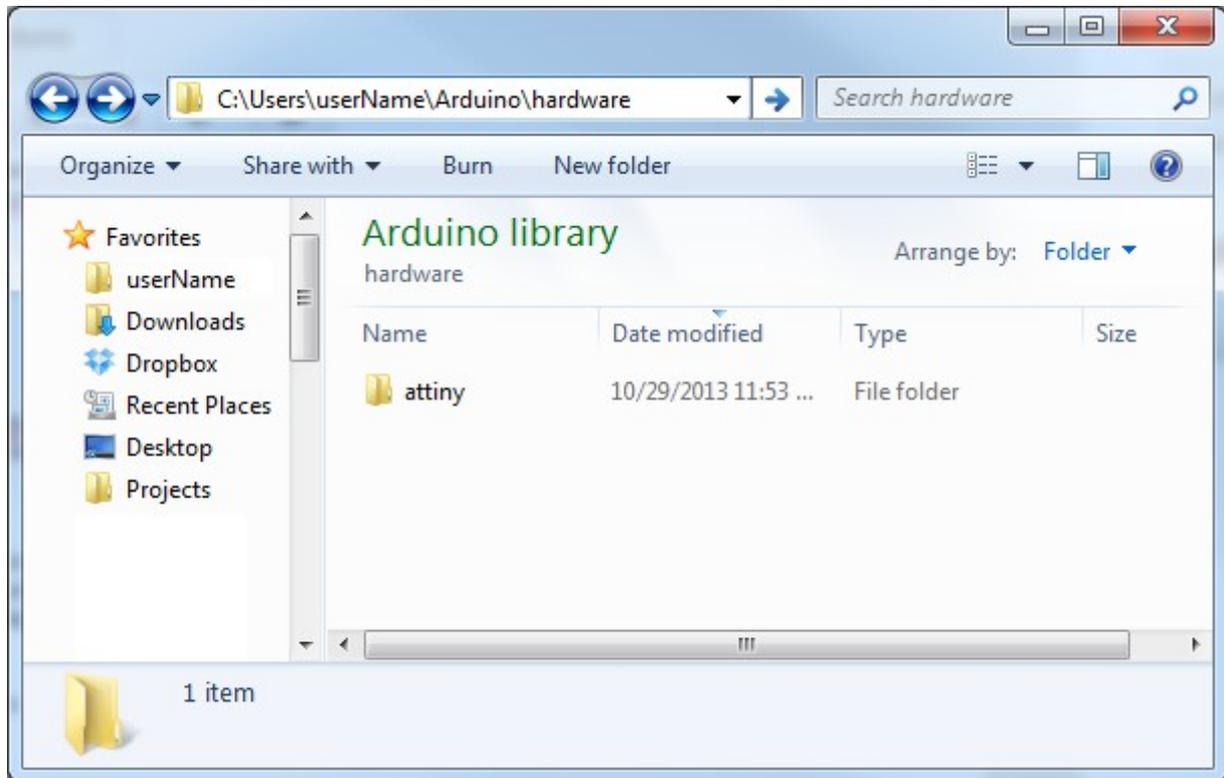
Déplacer le dossier attiny

Il devrait y avoir un dossier attiny dans le fichier attiny-ide-1.x.x.zip que vous avez téléchargé. Copiez ce dossier et collez-le dans un dossier appelé hardware dans votre répertoire Arduino Sketchbook.

Si vous ne savez pas où se trouve votre carnet de croquis Arduino, ouvrez Arduino et allez dans

Fichier > Préférences. L'emplacement du carnet de croquis doit être la première entrée de la boîte de dialogue Préférences. Par défaut, le carnet de croquis est généralement un dossier Arduino dans votre dossier personnel (par exemple, C:\Users\userName\Arduino sur Windows, ou /Users/userName/Documents/Arduino sur Mac).

S'il n'y a pas de dossier hardware dans votre Sketchbook, créez-en un. Après y avoir placé le dossier attiny, votre structure de répertoire devrait ressembler à ceci :

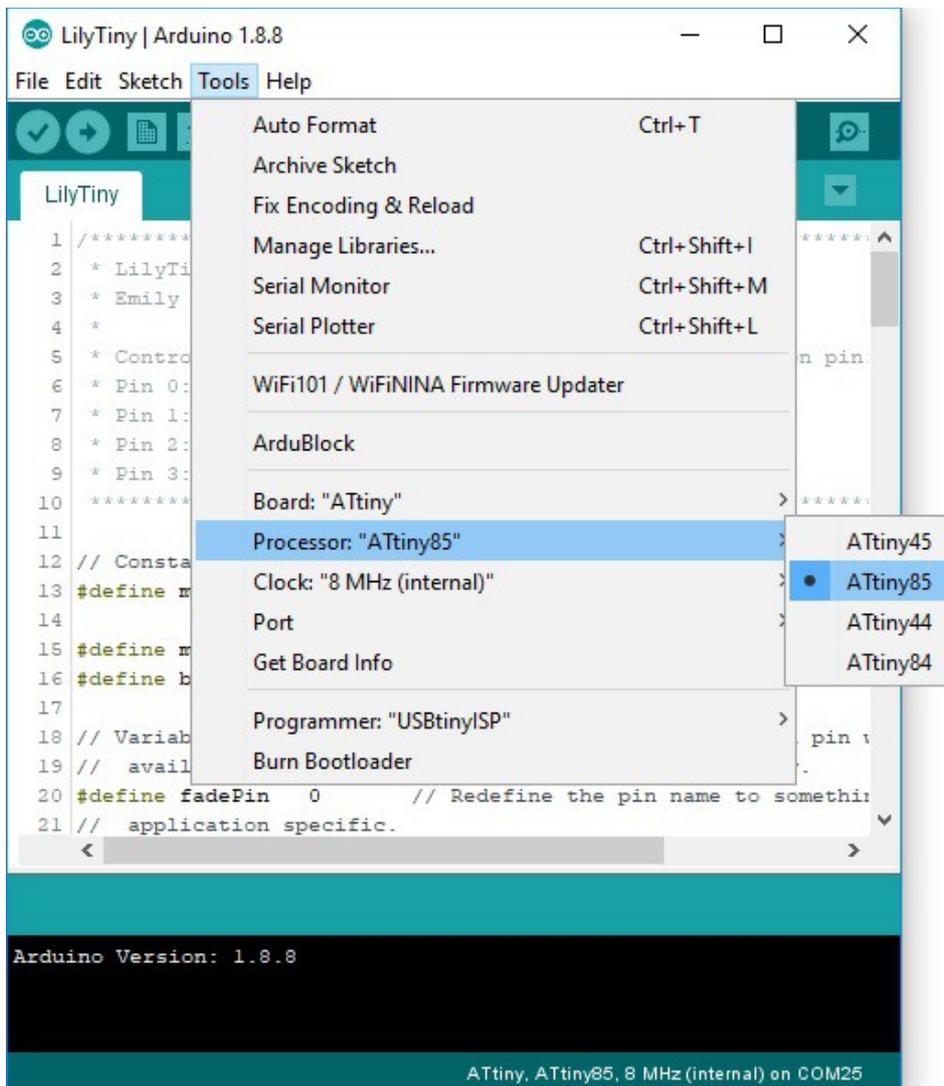


Ouvrir et configurer Arduino

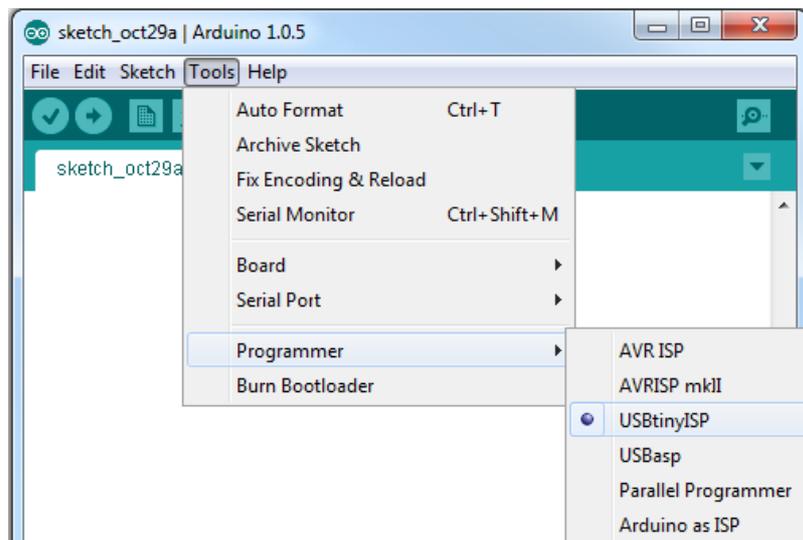
Nous arrivons bientôt à la partie la plus amusante ! Ouvrez Arduino. Si vous avez ouvert Arduino à la dernière étape, fermez-le et redémarrez-le.

Dans le menu Tools > Board, vous trouverez les effets du dossier attiny. Il devrait y avoir douze nouvelles entités dans la liste des cartes, qui vous permettent de programmer les ATtiny45, 85, 44 et 84. Chaque microcontrôleur peut être réglé sur une variété de vitesses d'horloge -- interne 1MHz ou 8MHz ou externe 20MHz.

Si vous utilisez un ATtiny85 nu, qui n'a jamais été modifié, sélectionnez **ATtiny85 (horloge interne de 1 MHz)**. Faites attention en choisissant ici, l'option 8 MHz ne fera que ralentir votre sketch, mais l'option 20 MHz peut " bloquer " votre ATtiny. **Ne sélectionnez pas l'option 20 MHz à moins d'avoir une horloge externe connectée !**

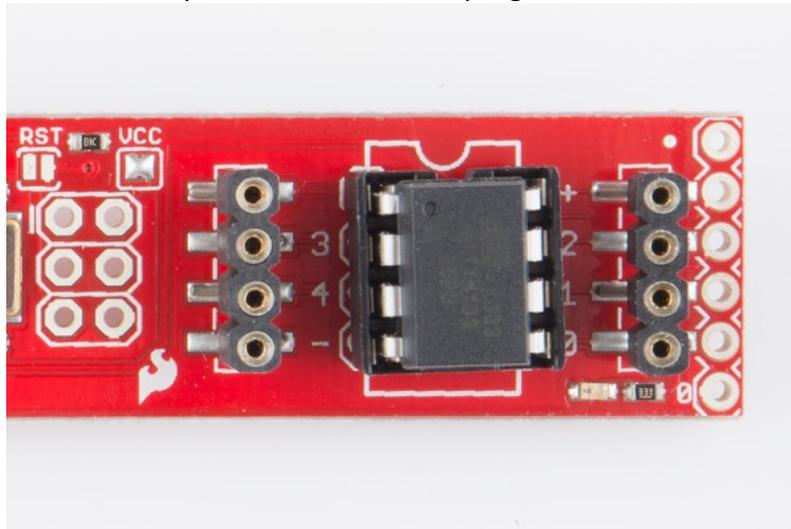


Contrairement aux autres cartes Arduino, vous n'avez pas besoin de sélectionner un port série lorsque vous utilisez le Tiny AVR Programmer. Mais vous devez sélectionner un programmeur. Dans le menu Outils > Programmeur, sélectionnez USBtinyISP.



Installer l'ATtiny

On s'approche du clignotement ! Lorsque vous installez l'ATtiny dans votre programmeur, assurez-vous que sa position de la polarité est correcte. Le petit cercle gravé sur le circuit intégré doit s'aligner sur l'encoche de la prise et de l'écran du programmeur.



Pour insérer le circuit intégré dans le socle, il peut être nécessaire de plier légèrement les pattes de chaque côté vers l'intérieur.

Télécharger le code !

C'est l'heure du sketch Blink ! Le Tiny AVR Programmer possède une LED intégrée, connectée à l'ATtiny, que nous pouvons utiliser pour vérifier que le code sur le circuit intégré est en cours d'exécution. La LED est connectée à la broche 0 dans l'environnement Arduino. Copiez/collez ce code dans votre fenêtre Arduino :

```
int blinkPin = 0;

void setup()
{
  pinMode(blinkPin, OUTPUT);
}

void loop()
{
  digitalWrite(blinkPin, HIGH);
  delay(500);
  digitalWrite(blinkPin, LOW);
  delay(500);
}
```

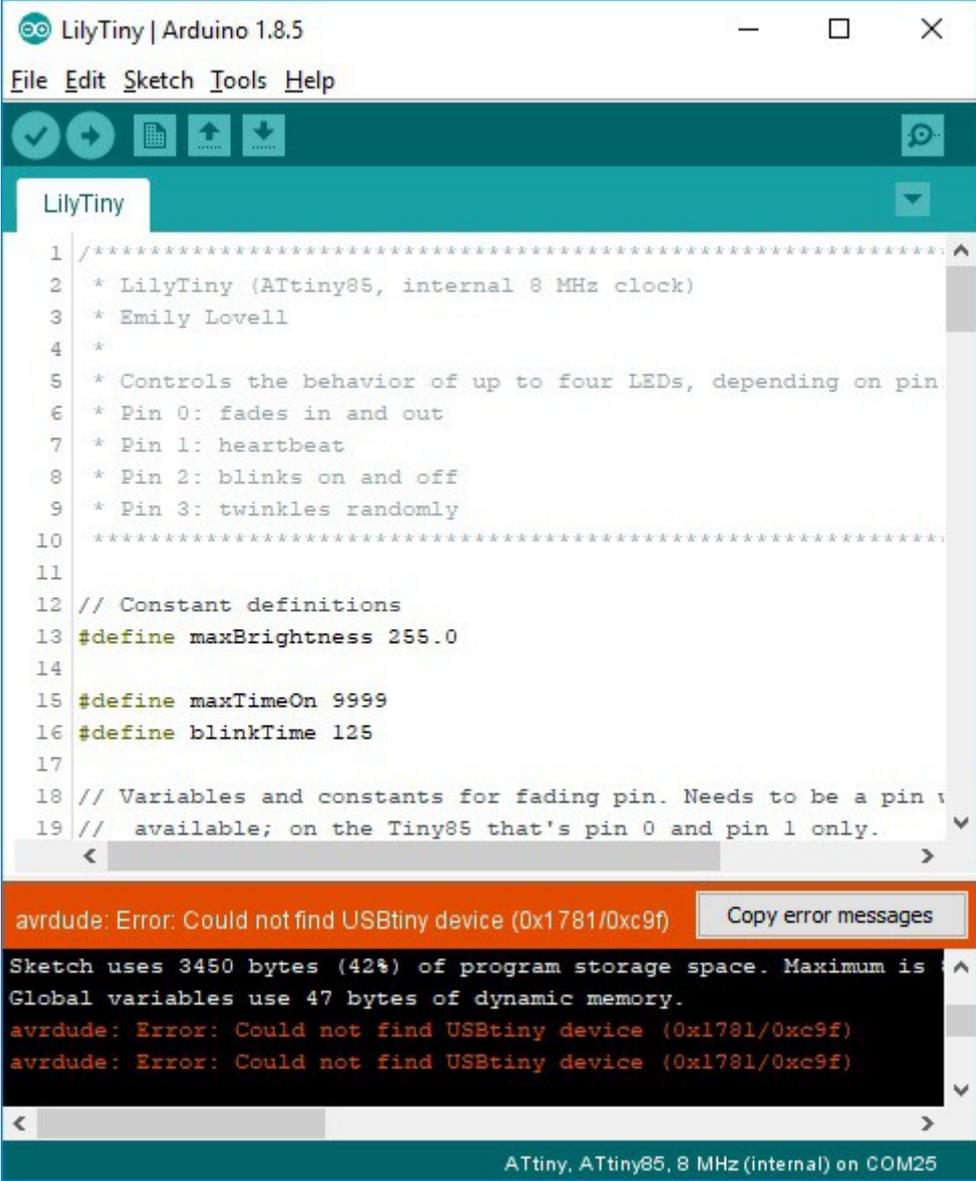
Cliquez ensuite sur le bouton Upload comme vous le feriez avec n'importe quelle carte Arduino. Le code sera compilé, et il devrait être téléchargé très rapidement. Ce sont les merveilles de la programmation directe dans le système. En cas de succès, la LED ambre intégrée devrait commencer à clignoter.

Note : Vous pouvez également télécharger en utilisant le menu de l'Arduino IDE. Selon la version de votre Arduino IDE, vous pouvez sélectionner Sketch > Upload Using Programmer ou Sketch > Upload Using Programmer.

☐ Conseil de dépannage : Si vous recevez l'erreur Arduino suivante, cela peut être dû à la connexion ou à un problème avec les pilotes.

```
avrdude: Error: Could not find USBtiny device (0x1781/0xc9f)
```

Vous devriez voir ceci en bas de l'IDE Arduino dans la console de texte.



The screenshot shows the Arduino IDE interface. The top window is titled 'LilyTiny | Arduino 1.8.5'. Below the menu bar, there are icons for running, uploading, and downloading. The main editor area shows the following code:

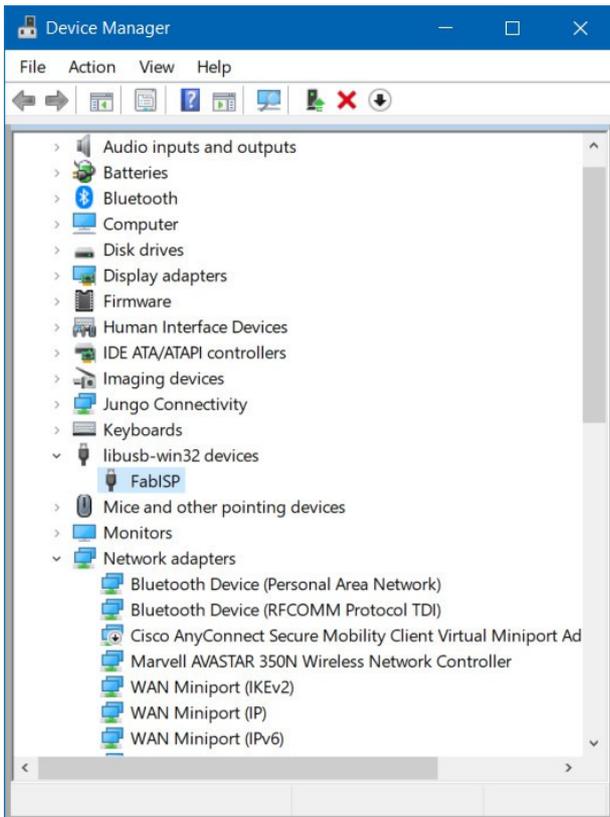
```
1 /*****  
2  * LilyTiny (ATtiny85, internal 8 MHz clock)  
3  * Emily Lovell  
4  *  
5  * Controls the behavior of up to four LEDs, depending on pin  
6  * Pin 0: fades in and out  
7  * Pin 1: heartbeat  
8  * Pin 2: blinks on and off  
9  * Pin 3: twinkles randomly  
10 *****/  
11  
12 // Constant definitions  
13 #define maxBrightness 255.0  
14  
15 #define maxTimeOn 9999  
16 #define blinkTime 125  
17  
18 // Variables and constants for fading pin. Needs to be a pin v  
19 // available; on the Tiny85 that's pin 0 and pin 1 only.
```

The console window at the bottom shows the error message: `avrdude: Error: Could not find USBtiny device (0x1781/0xc9f)`. A button labeled 'Copy error messages' is visible next to the error. Below the error, the console also displays: `Sketch uses 3450 bytes (42%) of program storage space. Maximum is` and `Global variables use 47 bytes of dynamic memory.` The status bar at the bottom indicates: `ATtiny, ATtiny85, 8 MHz (internal) on COM25`.

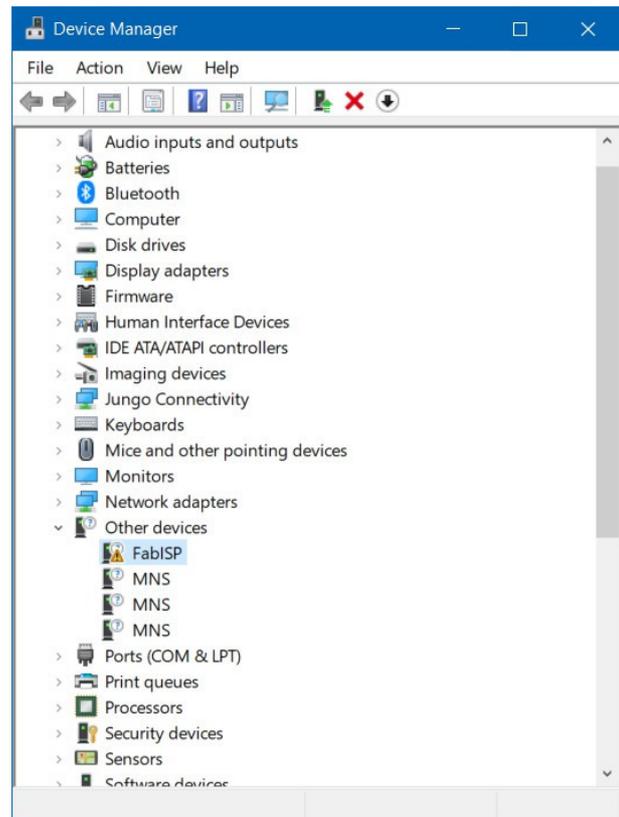
Si le problème est lié à la connexion, essayez de débrancher/rembrancher le programmeur AVR dans votre câble USB ou port USB. Vous pouvez également essayer un autre câble USB.

Dans le cas contraire, le problème peut être dû aux pilotes. Cela peut être dû au fait que le pilote n'a pas été installé correctement ou qu'il y a un conflit de pilotes. Ouvrez votre gestionnaire de périphériques pour afficher le périphérique. L'image de gauche montre que le périphérique

s'affiche sous libusb-win32 devices > FabISP. L'image de droite montre le périphérique sous Autres périphériques > FabISP.

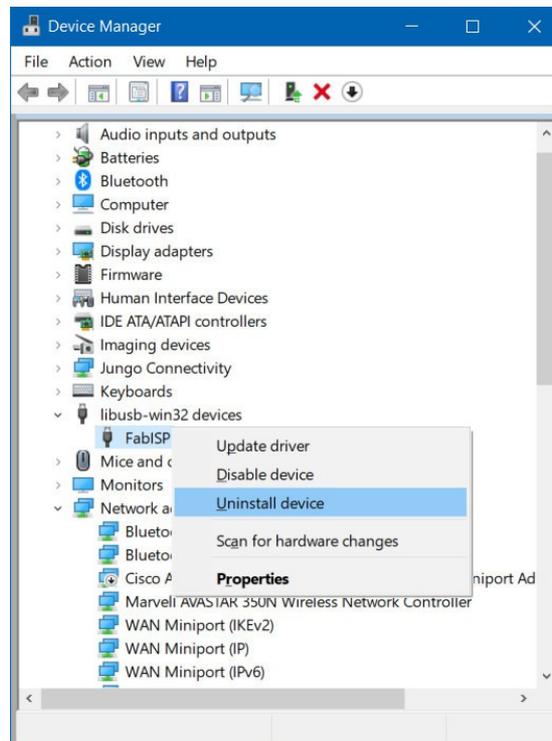


Pilote en conflit



Pilote non installé

Dans les deux cas, faites un clic droit sur le périphérique et sélectionnez Désinstaller le périphérique.



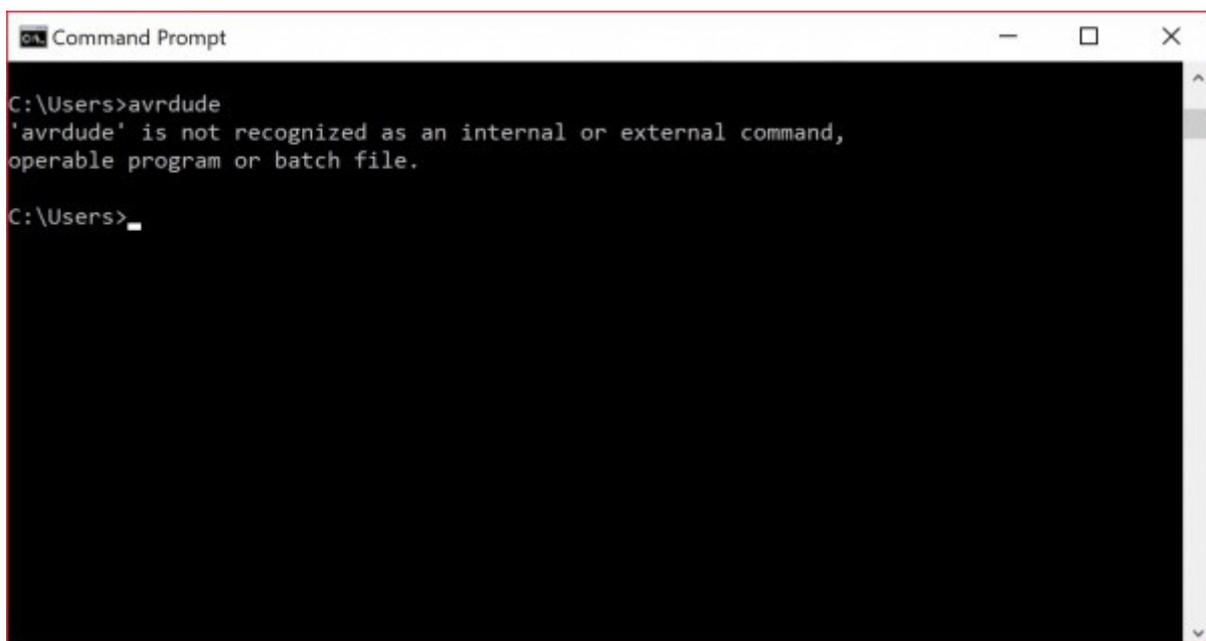
Une fenêtre similaire à l'image ci-dessous peut apparaître. Cliquez sur le bouton intitulé Désinstaller. Dans certains cas, Windows peut proposer l'option "Supprimer le pilote de ce périphérique" ; si c'est le cas, cochez simplement la case avant de cliquer sur le bouton de désinstallation.



Débranchez et rebranchez le programmeur AVR sur le port USB de votre ordinateur. Retournez à la section Installation des pilotes et suivez les instructions pour [installer automatiquement les pilotes à l'aide de Zadig](#).

Téléchargement de code à la dure

Si vous souhaitez avoir plus de contrôle sur votre Tiny AVR Programmer - et sur l'AVR auquel il est connecté - suivez le tutoriel pour le Pocket AVR Programmer. Bien que le tutoriel ait été écrit pour le Pocket AVR Programmer, il est équivalent sur le plan fonctionnel pour le Tiny AVR Programmer. Veillez simplement à vous connecter aux broches ICSP respectives de la puce AVR cible.



```
C:\Users>avrdude
'avrdude' is not recognized as an internal or external command,
operable program or batch file.
C:\Users>
```

[Utiliser AVRDUDE en ligne de commande](#)

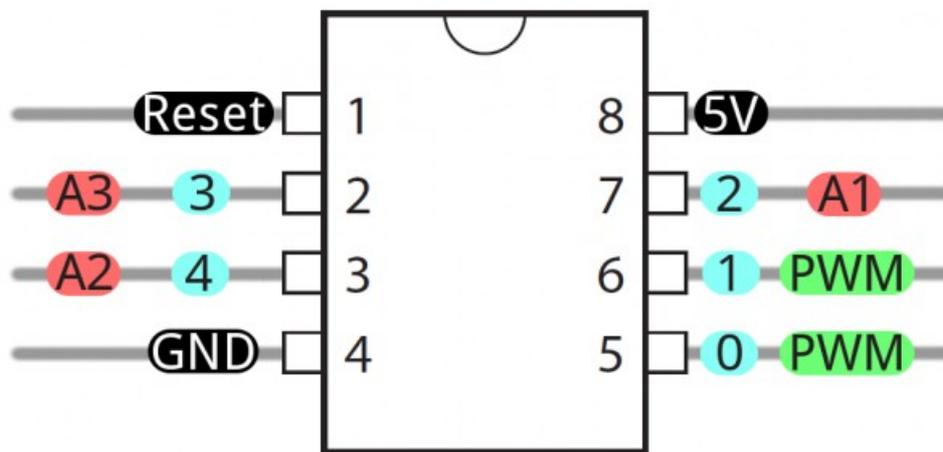
Conseils d'utilisation de l'ATtiny85

L'ATtiny85 n'est pas le circuit intégré Arduino de tous les jours. Il a beaucoup de punch pour sa petite taille, mais il y a certaines choses qu'il ne peut pas faire.

Sur cette page, nous allons donner un aperçu rapide de l'ATtiny85 en ce qui concerne Arduino et le Tiny AVR Programmer.

Brochage

Comme sur toute carte Arduino, chaque broche E/S de l'ATtiny85 est associée à un identifiant numérique. Ces broches sont documentées sur la carte, mais vous pouvez aussi vous référer à l'image ci-dessous en cas d'oubli.



Chacune des broches E/S de l'ATtiny85 est capable d'effectuer des entrées et sorties numériques. De plus, certaines broches ont des fonctionnalités spéciales.

Entrées et sorties analogiques

Il y a deux sorties analogiques et trois entrées analogiques. Utilisez-les comme vous le feriez avec n'importe quelle carte Arduino. Utilisez `analogWrite([pin], [0-255])` pour obtenir une sortie PWM. Cette fonctionnalité est disponible sur les broches 0 et 1. Par exemple :

```
int pwmPin = 0;

pinMode(pwmPin, OUTPUT);

for (int i=0; i<=255; i+=5)
{
  analogWrite(pwmPin, i);
  delay(5);
}
```

Et utilisez `analogRead([pin])` pour lire une tension analogique entre 0 et 5V, et la transformer en une représentation de 10 bits de cette tension. Les broches 2, 3 et 4 sont capables de recevoir une entrée analogique, mais lorsqu'elles sont utilisées en tant que telles, elles doivent être référencées comme **A1**, **A3** ou **A2** respectivement.

Par exemple :

```
int pwmPin = 0;
int analogInPin = A1;

pinMode(pwmPin, OUTPUT);
pinMode(analogInPin, INPUT);

int analogIn = analogRead(analogInPin); // Read analog voltage on pin 2 (A1)

analogWrite(pwmPin, analogIn / 4); // Output analog reading to dimmable LED
```

Note : Pour les utilisateurs avancés, vous pouvez modifier les registres timer/compteur de l'ATtiny85 pour augmenter le nombre de canaux PWM disponibles ! Vous pouvez obtenir jusqu'à quatre canaux PWM en suivant ce tutoriel ici : [Technoblogy - Four PWM Outputs from the ATtiny85](#) (en anglais). L'exemple ne fait qu'exécuter le PWM sur l'ATtiny85. Si vous essayez d'ajouter d'autres fonctionnalités en dehors de l'exemple PWM, il pourrait y avoir des comportements inattendus puisque cela modifie les registres de synchronisation.

Si vous avez besoin de réinitialiser la puce, utilisez simplement la commande erase avec le Tiny AVR ou le Pocket AVR Programmer via la ligne de commande pour la ramener à son état précédent. Charger du code avec l'IDE Arduino ne sera pas suffisant. Voici un exemple utilisant les réglages des bits de fusible pour l'ATtiny85 de LilyTwinkle :

```
avrdude -c usbtiny -b 19200 -p t85 -v -e -U lfuse:w:0xe2:m -U
hfuse:w:0xdf:m -U efuse:w:0xff:m -U lock:w:0xCF:m
```

Après avoir réinitialisé la puce, vous pouvez télécharger le code dans la puce via l'IDE Arduino comme expliqué précédemment.

Aucune liaison série (UART). Oui SPI et I2C

Vous pouvez remarquer, sur la liste des fonctions des broches spéciales, qu'il n'y a pas de RX ou TX UART. C'est parce que l'ATtiny85 n'a pas d'UART intégré. Si vous essayez de compiler un code Arduino avec `Serial.begin(9600)`'s ou `Serial.print()`'s vous obtiendrez une erreur.

Vous êtes donc privé de l'un des outils de débogage Arduino les plus utiles. Vous ne pouvez pas imprimer sur le Serial Monitor. Mais l'ATtiny85 a toujours [I2C](#) et [SPI](#), qui sont beaucoup plus couramment utilisés pour la communication des capteurs de nos jours. Malheureusement, les bibliothèques Arduino pour ces interfaces n'ont pas encore été écrites pour l'ATtiny85, mais il y a quelques bibliothèques fournies par les utilisateurs sur le web. [USI2c](#) est une bibliothèque Arduino qui active I2C sur l'ATtiny85.

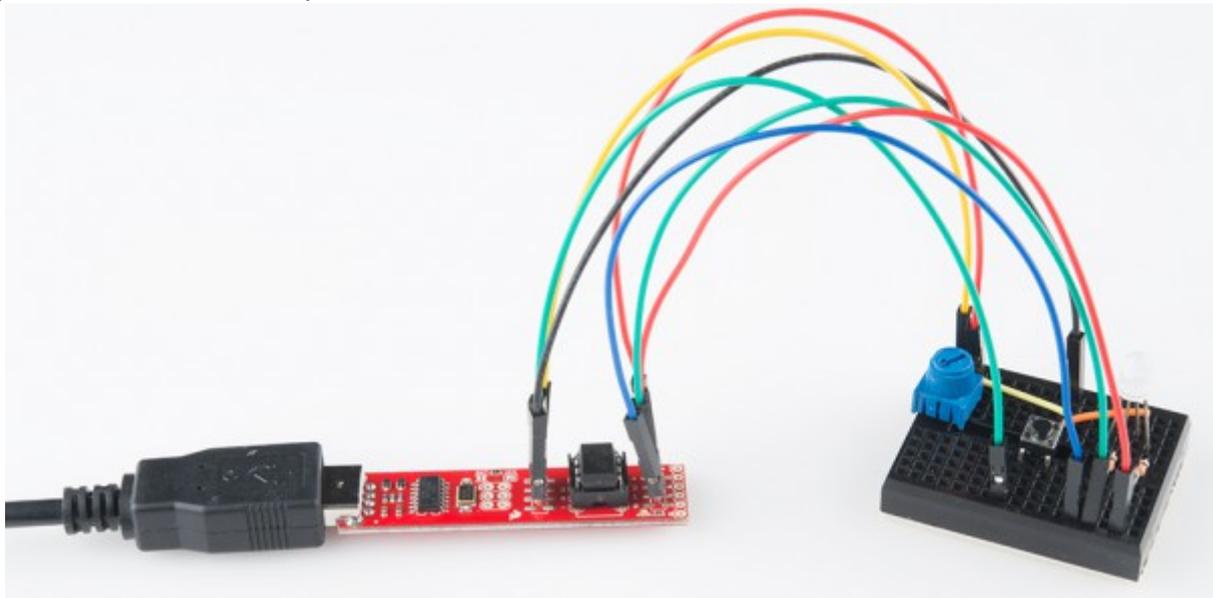
Il existe également d'autres bibliothèques centrées sur l'ATtiny85. Comme [Servo8Bit](#), une bibliothèque de servo.

Astuce : Vous cherchez un guide de référence rapide pour l'ATtiny85 ? Cliquez sur [ce lien](#) pour télécharger une image ou une version PDF de nos ressources !

Prototypage avec le Tiny AVR Programmer

Il n'y a pas de limite à l'excitation que l'on peut ressentir en faisant varier la luminosité d'une simple LED jaune. Vous finirez par vouloir aller plus loin et commencer à connecter votre tiny85 à d'autres composants électroniques. Il y a plusieurs façons de le faire.

La méthode de prototypage la plus simple et la moins permanente consiste à utiliser les connecteurs de prototypage situés de part et d'autre de la prise. Vous pouvez connecter des fils de connexion mâles standard à ces broches, qui peuvent à leur tour être acheminés vers des circuits imprimés ou d'autres composants.



Pour les projets plus permanents, il est assez facile de retirer délicatement le circuit intégré de son socle et de le brancher sur un circuit imprimé ou une "breadboard". Finalement, une fois que vous aurez suffisamment travaillé sur votre projet, c'est probablement là que vous voudrez aller.

Boîtiers SOIC pour ATtiny85 montés en surface (CMS)

Vous essayez de reprogrammer un ATtiny85 avec un boîtier SOIC ? Il y a plusieurs façons de se connecter. La plus simple est d'utiliser le clip de test IC et les fils de liaison M/F.

Ressources et Aller plus loin

Maintenant que vous avez réussi à faire fonctionner votre Tiny AVR Programmer, il est temps de l'incorporer dans votre propre projet !

Pour plus d'informations, consultez les ressources ci-dessous :

Fichiers de conception du Tiny AVR Programmer

- [Schéma](#) -- Un PDF du schéma du Tiny AVR Programmer.
- [Eagle Files](#) -- Si vous voulez regarder le design du PCB, ou le modifier pour en faire une version personnelle, regardez ces fichiers.

- [Tiny AVR Programmer Firmware](#) -- Si vous voulez creuser dans le code qui vit sur le programmeur lui-même, regardez ceci.
- [Dépôt GitHub](#) -- Allez ici pour trouver la dernière et meilleure version du matériel et du firmware du Tiny AVR Programmer. Ou modifiez-le et apportez vos modifications !

Pilotes, etc.

- [Définitions de la carte Arduino](#) -- Le dossier attiny doit se trouver dans un dossier hardware dans votre sketchbook Arduino.
- [Logiciel Zadig v2.0.1.160 et USBtiny \(ZIP\)](#) -- Pour une installation automatique.
- [Pilote USBTinyISP 32-bit](#) -- Pilote Windows pour une installation manuelle sur les systèmes 32-bit
- [Pilote USBTinyISP 64-bit](#) -- Pilote Windows pour installation manuelle sur systèmes 64-bit
- [GitHub : Signed USBTiny Drivers](#) -- Si les 3 premières options listées dans le tutoriel ne s'installent pas ne s'installent pas.

Ressources pour Attiny85

- [Atmel Documentation Page](#) -- Les dernières fiches techniques et notes d'application pour l'ATtiny85.
- [High-Low Tech Tutorial](#) -- Une vue d'ensemble de la programmation de l'ATtiny85 en utilisant Arduino.
- [Guide de référence](#) rapide pour l'ATTiny85

Besoin d'inspiration pour votre prochain projet ? Jetez un coup d'œil à ces tutoriels :

- [H2OhNo !](#) -- La carte de développement et d'alarme d'eau H2OhNo ! utilise un ATtiny85 pour détecter la présence d'eau. Ce tutoriel va plus loin dans le passage de l'ATtiny85 en mode très basse consommation.
- [Registres à décalage](#) (shift register)-- Si vous vous sentez limité par le manque de broches de l'ATtiny, vous pouvez utiliser un registre à décalage pour augmenter le nombre d'entrées/sorties.
- [Utilisation de l'Arduino Pro Mini 3.3V](#) -- Si vous cherchez un petit boîtier, mais que vous avez besoin de plus de broches et de fonctionnalités, jetez un coup d'oeil à l'Arduino Pro Mini.
- [Installation d'un chargeur de démarrage Arduino](#) -- Vous pouvez utiliser le Tiny AVR Programmer pour programmer toutes sortes d'AVR, y compris ceux qui se trouvent sur la plupart des cartes compatibles avec Arduino. Si vous avez besoin de reprogrammer votre chargeur de démarrage Arduino, le Tiny AVR Programmer devrait être tout ce dont vous avez besoin.
- [Pocket AVR Programmer Hookup Guide](#) -- Si vous souhaitez utiliser le Tiny AVR Programmer pour programmer d'autres AVR avec AVRDUDE via la ligne de commande, ce tutoriel fournira quelques exemples et des conseils de dépannage. Veillez simplement à vous connecter aux broches ICSP correspondantes.

Achat direct du programmeur en France : [GO TRONIC](#) avec son premier [TTINY85-20PU](#)