

Programmer un ATtiny85 à l'aide d'une carte Arduino Uno

Présentation de l'ATtiny85

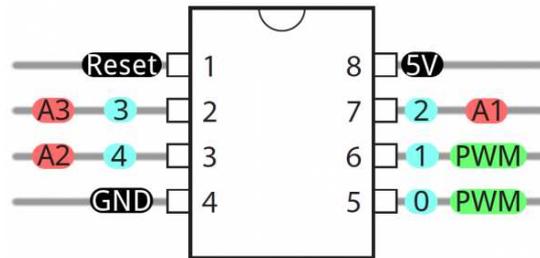


Fig. 1 - pinout de l'ATtiny85 ([source de l'image](#)).

Le microcontrôleur dispose de 5 broches digitales ; 3 broches analogiques ; 2 broches de PWM.

Préparation de la carte Arduino Uno

- Connecter la carte Arduino Uno au PC, lancer l'IDE et vérifier que le type de carte et le port sélectionnés soient corrects.
- Outils / programmeur... / Arduino as ISP
- Fichier / exemples / arduinoISP
- Téléverser le programme et débrancher la carte

Câblage de l'ATtiny85 à la carte Arduino

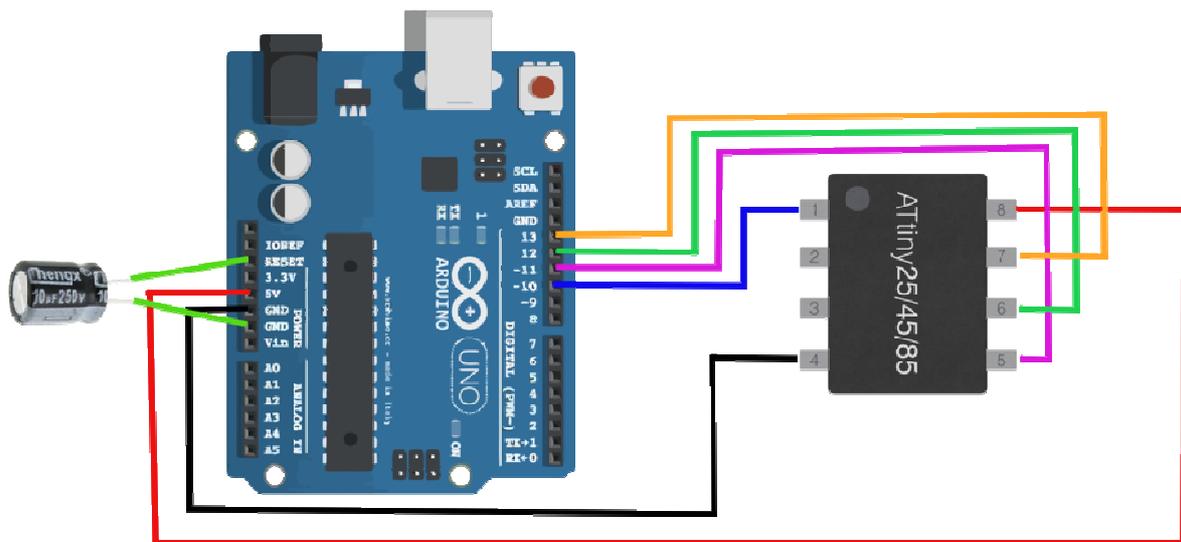
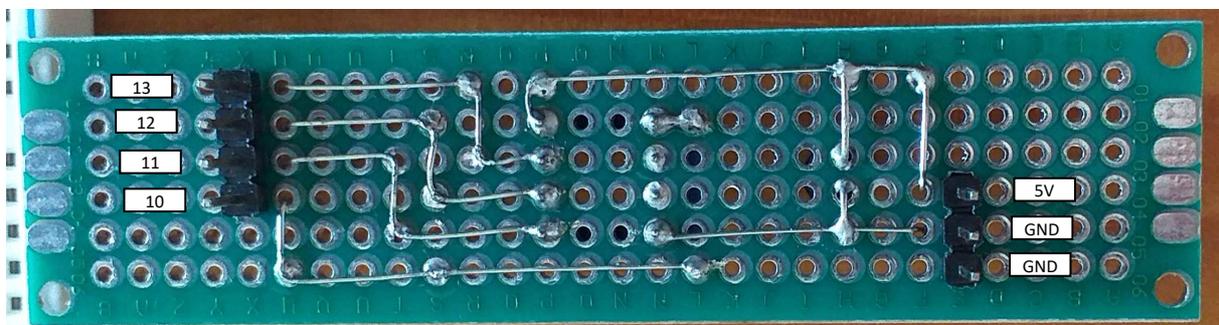
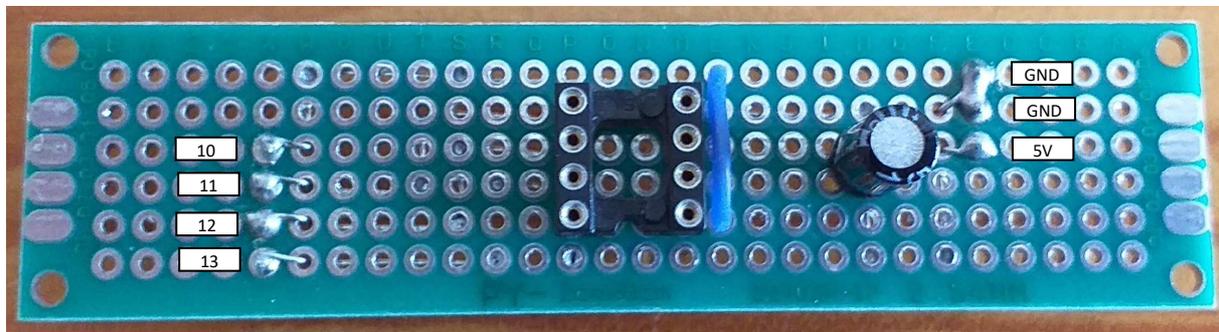


Fig. 2 – câblage de l'ATtiny85 à la carte Arduino [\(source de l'image\)](#).

On remarquera l'ajout d'un condensateur de 10 uF entre les bornes Reset et GND / ou entre 5V et GND... je pense que le rôle de ce condensateur est de palier une chute de tension intempestive de l'alimentation, ce qui relancerait le programme de façon non désirée.

Réalisation d'un « shield »



Programmation de l'ATtiny85

- Fichier / Préférences / gestionnaire de carte :

https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json

- Outils / Type de carte / gestionnaire de carte

Installer la bibliothèque : attiny by David A. Mellis version 1.0.2 (le 17/04/2022)

- Outils / Sélectionner le bon type de carte : ATtiny microcontrollers... => ATtiny85
- Outils / processeur : ATtiny85
- Outils / clocks : 8MHz

Essai programme

Ce programme a pour but de vérifier le bon fonctionnement du circuit intégré... cinq DEL s'allument les unes après les autres (*variante de « blink »*).

```
void setup() {
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);

  digitalWrite(0, LOW);
  digitalWrite(1, LOW);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
}

void loop() {
  digitalWrite(4, LOW);
  digitalWrite(0, HIGH);
  delay(1000);

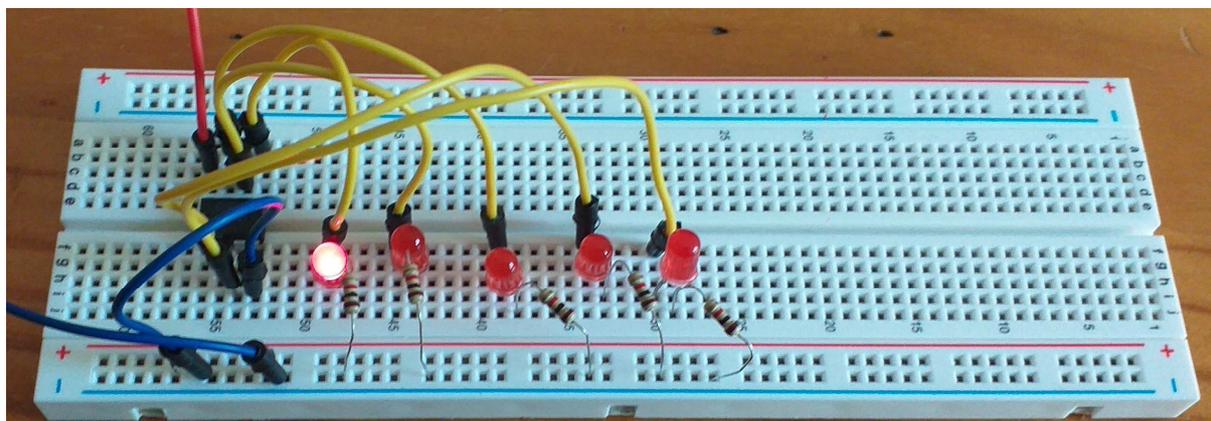
  digitalWrite(0, LOW);
  digitalWrite(1, HIGH);
  delay(1000);

  digitalWrite(1, LOW);
  digitalWrite(2, HIGH);
  delay(1000);

  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  delay(1000);

  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(1000);
}
```

Câblage de l'ATtiny85 dans un montage



- Connecter une DEL et une résistance de protection (1kohms) en série entre une sortie D0, D1, D2, D3 et/ou D4 (fils jaune) et GND (fils bleu).
- VCC doit être de 2,7 à 5,5 V (fils rouge). J'utilise une alimentation stabilisée mais une pile 4,5 V ou deux pile 1,5 V font l'affaire.
- Il est préférable de connecter la broche RESET, à travers une résistance de 10kohms, à la broche VCC (non présente sur la photo). Voir le DATASHEET paragraphe 2.2.4 page 4 : <https://pdf1.alldatasheet.fr/datasheet-pdf/view/174761/ATMEL/ATTINY85.html>
- Sinon ça fonctionne comme une carte Arduino...
- Remarque : Les durées ne sont pas calibrées : 1000 ms dans le code font approximativement 5 à 6 secondes en réalité... sûrement dû au fait que l'ATtiny85 fonctionne sur 8MHz.