

Table des matières

Introduction	7
Le matériel	9
Environnement de développement	11
Programmation de l'ATtiny	15
Lumière	17
Commande de LED	17
Chercheur de lumière	19
À la poursuite du soleil	22
Couleurs RVB	25
Étoile de fête	27
Chandelier	29
Sapin de Noël	30
Stroboscope pour soirée	31
Persistance rétinienne	32
Gyrophare	36
Registre à décalage	37
Scanneur KITT (<i>Knight Industries Two Thousand</i>).	38
Théâtre d'ombres	40
Scanneur KITT n° 2	41
Persistance rétinienne de puissance (<i>Power POV</i>)	42
Temporisation (555)	47
Minuterie de chambre noire	47
Générateur d'impulsions	48
Modulation de largeur d'impulsion	49
Minuteur pour la cuisson d'œufs	50
Minuterie pour insolation de circuit imprimé	52
Son	53
Métronome	53
Sirène	54
Veilleuse avec gradateur et détection de pleurs	54
Percussions électroniques	56

Carillon à vent	60
Carillons à vent à plusieurs tons	60
Interrupteur à claquement de mains	63
Détection	65
Détecteur d'intrusion à infrarouge passif	65
Photographie TGV	67
Sonnette sensitive	69
Indicateur de température	71
Ventilateur de bureau automatique	73
Détecteur de courrier	74
Détecteur lumineux	76
Moteurs pas-à-pas	79
Commande des moteurs pas-à-pas	79
Robot copieur	82
Robot dessinateur	87
Dessinateur	89
Visualisation sur 360°	90
Petits jeux	93
Thérémine	93
Dé	94
Match 3	98
Divers	101
Circuit imprimé universel	101
Plate-forme de robot	102
Pendule à mouvement perpétuel	105
Régulateur de vitesse	108
Générateur de fonctions	109
Conclusion	111

Introduction

Si vous connaissez les cartes à microcontrôleur modernes, comme l'Arduino, vous êtes aussi familier de leurs limitations : trop chères pour en utiliser une avec chaque projet, même si on déniche une version bon marché « *made in China* », et pas si petites que ça, bien que la Nano soit un pas dans la bonne direction. Le nombre élevé de broches d'entrées/sorties complique en outre le dessin d'un circuit imprimé pour les extensions.

Un petit microcontrôleur de la taille d'un 555 et facile à programmer, serait l'idéal. Il en existe un ! L'ATtiny85 possède une mémoire interne de 8 KB et coûte à peine plus d'un euro, c'est parfait pour les débutants. Il existe aussi un modèle avec 4 KB de mémoire, l'ATtiny45 ; la plupart des projets de ce livre s'en contenteraient, et son prix est sensiblement le même.

La taille de l'ATtiny est trompeuse : on pourrait croire que ses possibilités sont limitées, mais il possède un nombre surprenant de caractéristiques intéressantes (et ludiques). Ce livre exploite les capacités de ce circuit intégré dans une cinquantaine de projets électroniques, spécialement conçus pour un amateur débutant. La plupart sont traités de manière exhaustive, quelques-uns ne proposent qu'une ébauche pour un développement ultérieur. La table des matières montre la diversité des domaines abordés.

Ce livre n'est pas un manuel théorique, et il devrait être lu un fer à souder à la main. J'ai en effet horreur des platines d'expérimentation, et il est préférable d'utiliser des petits circuits imprimés (ceux de nos projets sont simples et peu coûteux à réaliser), ou à tout le moins des morceaux de plaques d'essai pastillées. Ceci ne doit pas vous empêcher d'utiliser une platine d'expérimentation dans la phase d'étude et de développement ; les circuits de ce livre s'y adaptent aisément.

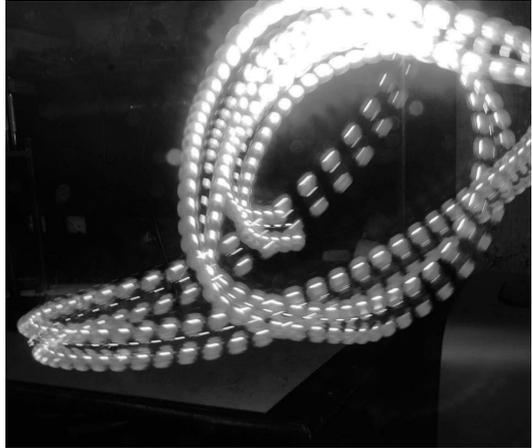
Pour programmer l'ATtiny, nous utilisons... l'Arduino, configuré en mode programmeur. Les scripts (*sketches*) de l'Arduino peuvent, pour la plupart d'entre eux, être utilisés avec l'ATtiny, mais certaines fonctions ne sont pas supportées telles quelles. Les premiers chapitres de ce livre donnent les explications nécessaires à ce sujet.

Ces aspects techniques sont sans doute le cap le plus difficile du livre, mais une fois ce pas franchi, les projets vous sembleront un jeu d'enfant.

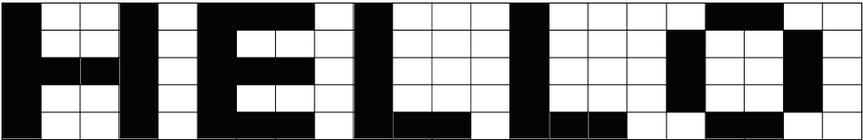
Pour terminer cette introduction, une remarque concernant l'alimentation. Nous indiquons en général 5 V, mais deux ou trois piles alcalines AA ou AAA suffisent en principe. Les montages sont donc portables, nul besoin d'un adaptateur secteur.

Lumière

```
//stroboscope
int leds= 0;
int Sup = 2;
int Sdwn = 3;
int up;
int dwn;
int del=200;
void setup() {
  pinMode(leds, OUTPUT);
}
void loop() {
  up=analogRead(Sup);
  dwn=analogRead(Sdwn);
  if(up<200){del=del-1;}
  if(dwn<200){del=del+1;}
  if(del<5){del=5;}
  digitalWrite(leds,HIGH);
  delay(del);
  digitalWrite(leds,LOW);
  delay(del);
}
```



Persistence rétinienne



Nous allons réaliser une expérience sur la persistance rétinienne (*POV – persistence of vision*), phénomène attribué à la rémanence des images sur la rétine de l'œil. Cela nous permet de percevoir comme continu et fluide un mouvement qui ne l'est pourtant pas. C'est le cas lorsque nous regardons la télévision, qui n'est jamais qu'une grosse lampe clignotante.

L'ATtiny, utilisé en temporisateur, nous permet de projeter un texte avec cinq LED. Le programme est sommaire : pas d'alphabet préprogrammé donc, mais tout simplement allumage et extinction des LED.

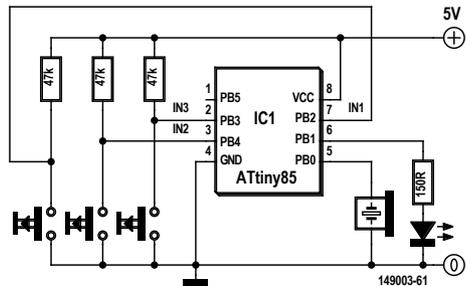
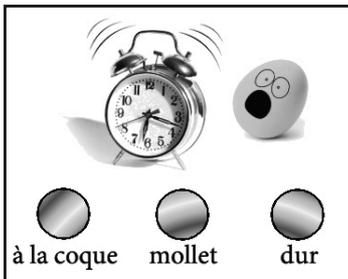
Le circuit est semblable à celui utilisé pour le sapin de Noël, et le circuit imprimé est simple. Le montage est alimenté par deux piles AAA. Il sera donc compact, ce qui permettra de le mettre en mouvement manuellement ou en le fixant sur un bras tournant.

Temporisation (555)

```
int Sortie = 0;
int t;
void setup() {
  pinMode(Sortie, OUTPUT);
}
void loop() {
  t=analogRead(Entrée); //Entre 0 et 1024
  analogWrite(Sortie, t/4); //Entre 0 et 255
}
```

```
//Soft Start
int Sortie = 0;
int x=0;
void setup() {
  pinMode(Sortie, OUTPUT);
}
void loop() {
  analogWrite(Sortie, x); //Entre 0 et 255
  x=x+1;
  if(x>255){x=0;}
  delay(10);
}
```

Minuteur pour la cuisson d'œufs



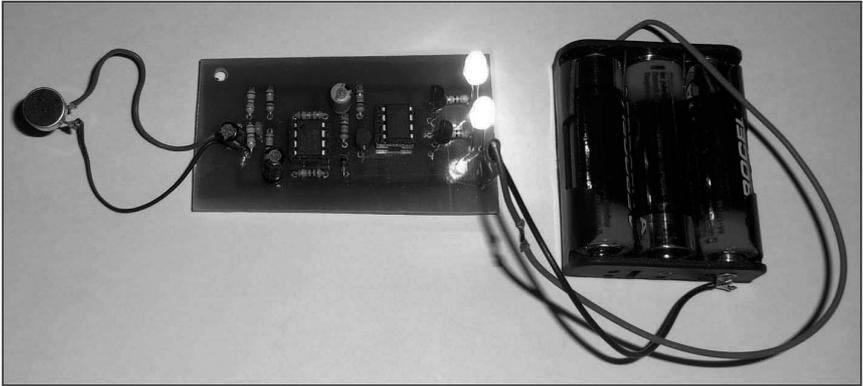
Considérant que l'eau bout lorsque nous y plongeons l'œuf, la durée de cuisson est d'environ 4 minutes pour un œuf à la coque, 6 pour un œuf mollet, et 8 pour un œuf dur. Ces durées peuvent bien entendu être modifiées en fonction de vos préférences personnelles.

Trois boutons-poussoirs, correspondant à ces modes de cuisson, et un bipleur (*buzzer*) font office respectivement d'entrées et de sortie. Une LED complète le schéma pour indiquer le début d'un cycle de cuisson ; elle s'allume dès qu'un bouton-poussoir est activé.

L'entrée 2 est à 4,25 V en l'absence de bruit, et à moins de 3 V lors de la détection de bruit. Pour une tension de 5 V, *AnalogRead* fournit une valeur de 1024 ; nous obtiendrons donc 870 en l'absence de bruit, 600 en présence de bruit. Le seuil de déclenchement est fixé par la valeur de l'entier *val*, comprise entre 600 et 870.

Pour réduire la sensibilité de l'amplificateur du microphone, il suffit de diminuer la valeur de la résistance de contre-réaction de 1 M Ω .

L'alimentation est assurée par trois piles AA. Les transistors MOSFET BS170 permettent de commander des LED de 20 ou 100 mA. On pourrait être tenté de mettre des LED en série, mais leur tension de déchet (c'est-à-dire le seuil de tension à partir duquel elles s'allument) dépasserait la tension d'alimentation. Et ne mettez jamais de LED en parallèle !



Percussions électroniques

Ici nous ne simulerons pas des bruits de tambours et cymbales, mais nous tambourinerons pour de vrai sur divers objets, comme des tasses ou des verres.

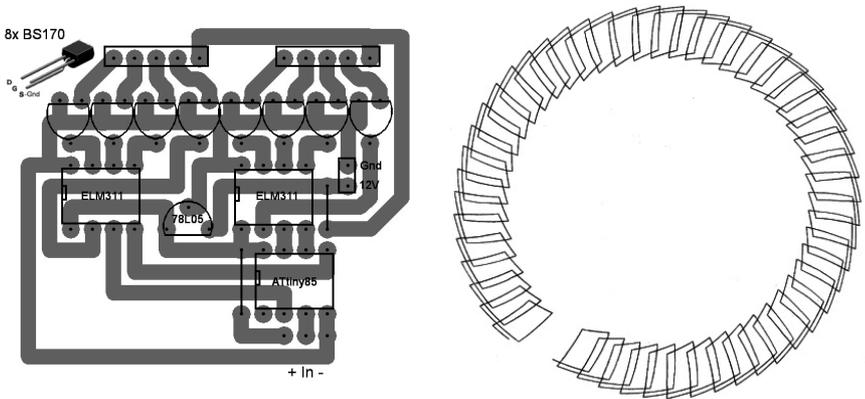
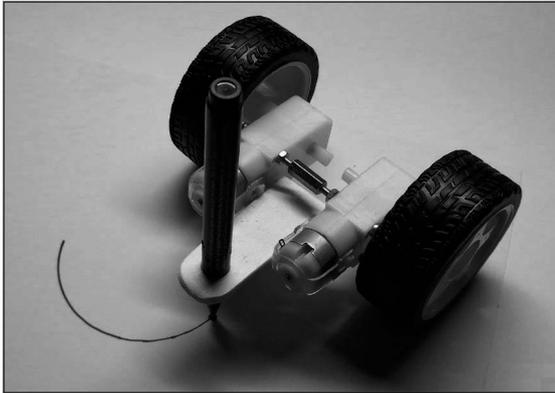
Pour animer ce montage, j'utilise un petit moteur à courant continu de récupération, alimenté sous 5 V, et sur l'axe duquel j'ai fixé avec les moyens du bord un écrou M3. Entraînée par son propre poids, une vis engagée dans cet écrou pend sous le moteur.

Lorsque nous alimentons le moteur, le boulon se balance et heurte tout ce qui se trouve sur son chemin. Coupons l'alimentation, et le boulon retourne pendouiller sous le moteur. Il n'est pas nécessaire que le couple du moteur soit suffisant pour faire tourner le boulon, un bon élan suffira.

Pour tambouriner sur divers objets, il suffit de brancher et couper l'alimentation du moteur en cadence. Le procédé est rudimentaire, mais il ne fait pas de doute qu'il s'agit bien de percussions.

Robot dessinateur

Si nous enlevons le capteur et le mécanisme de commande du marqueur du montage précédent, nous libérons une sortie de l'ATtiny et nous pouvons dès lors commander les deux moteurs séparément. Et il nous reste une entrée libre.



Le marqueur repose sur le papier (il faudra peut-être le lester), et le robot réalise de très beaux dessins.

Le *sketch* ci-dessous illustre les possibilités du robot : 2 000 pas dans une direction, puis 2 000 pas dans l'autre. La valeur du temps mort « d » dépend du moteur utilisé et de la vitesse souhaitée.