

# CHRISTMAS TREE TINY

Voici les premiers essais de mon nouveau Sapin de Noël programmé sur deux microcontrôleurs Amtel différents :

Le premier programme utilise Amtel Studio 6.1 et se loge facilement dans un Attiny 45 ou 85 dans mon test je n'utilise que 25 % de la mémoire de l'ATtiny45 car le but est de tout loger dans un ATtiny13 Le deuxième a été écrit pour Aduino 1.05 (il fonctionne aussi sur Arduino 1.04) pour le téléverser (Uploader) il faut ajouter cette librairie

à télécharger ici : <http://sourceforge.net/projects/ard-core13/>

Il y a d'autres supports pour les ATTiny voir le lien pour les « Boîtes de Chocolats à LED » à la fin de la page.

## Voici les fonctions prises en charge:

millis ()

micros () ( pas vraiment un vrai microseconde )

retard ()

delayMicroseconds () ( pas vraiment un vrai microseconde )

analogRead ()

analogWrite ()

pinMode ()

digitalRead ()

digitalWrite ()

Gardez cet ensemble de fonctions à l'esprit lors de l'écriture de votre logiciel - des fonctions plus complexes telles que les fonctions mathématiques et trigonométriques peuvent fonctionner mais c'est sans garantie pour un si petit microcontrôleur

## INSTALLATION DE LA LIBRAIRIE core13 :

voici l'exemple pour smeezekitty et son ZIP pour core13

Arrêtez l'IDE Arduino

Identifier le répertoire dans Arduino ( Sous Windows 7 dans mon ordinateur, le chemin est le suivant:

**C:\Program Files (x86)\Atmel\arduino-1.0.5\hardware\arduino\cores\**

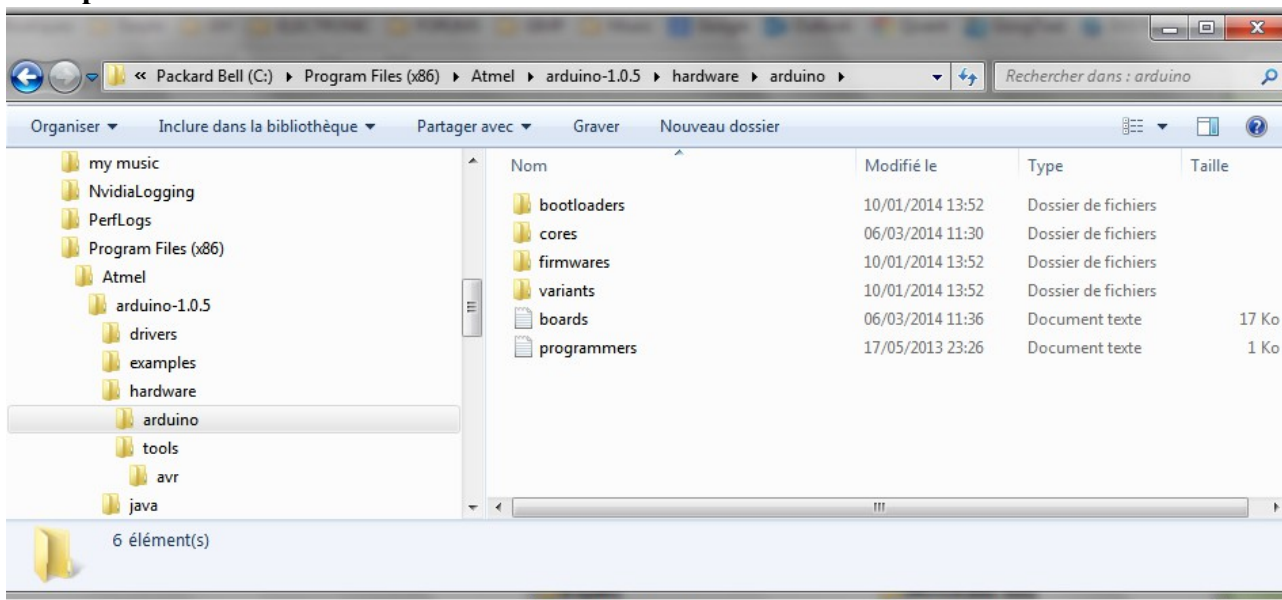
il suffit de dézipper le fichier dans le répertoire cores et un nouveau répertoire nommé core13 sera créé avec tous les fichiers nécessaires.

Si ce répertoire n'est pas créé par le ZIP il faut le créer et y dézipper les fichiers comme ici:

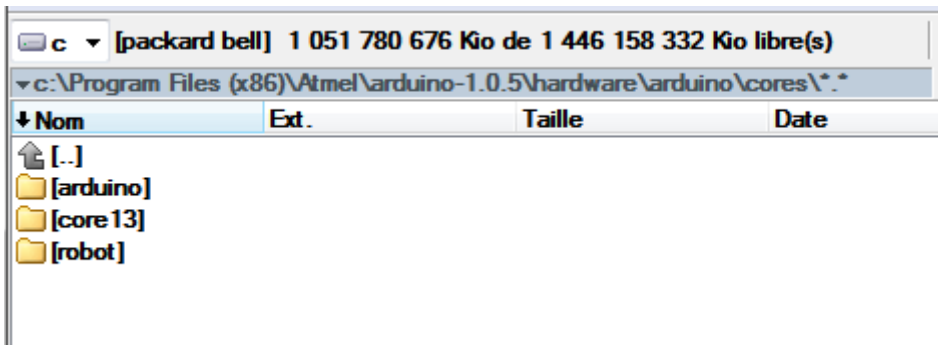
**C:\Program Files (x86)\Atmel\arduino-1.0.5\hardware\arduino\cores\core13\**

voici les captures d'écran :

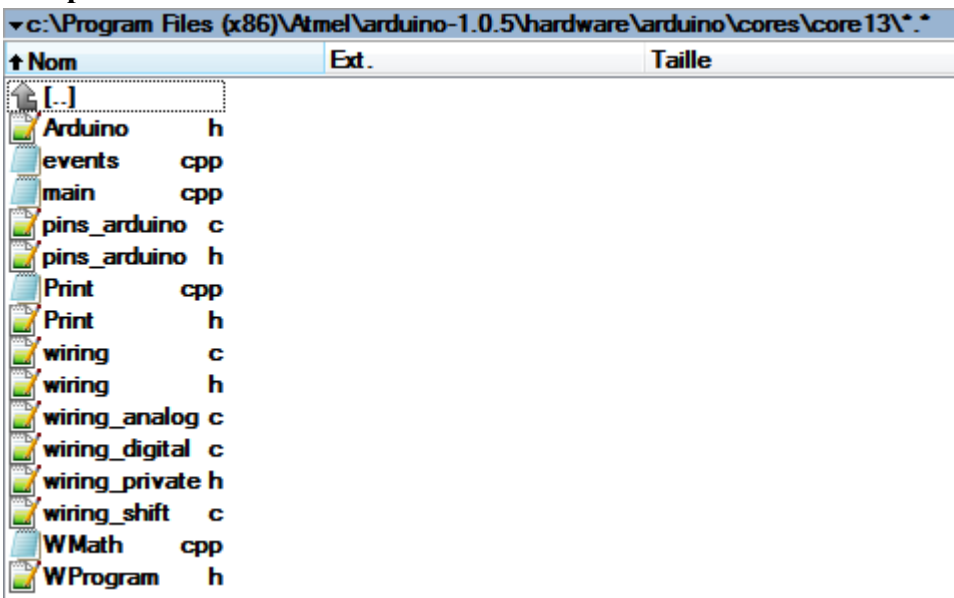
## du répertoire arduino



## du répertoire cores



## du répertoire core13



dans le répertoire `arduino-1.0.5\hardware\arduino\` il y a un fichier texte nommé `boards.txt`, pour ATtiny13 il faut insérer ces lignes à la fin du fichier :

```
#####
```

```
attiny13.name=Attiny13 @ 128 KHz (internal watchdog oscillator)
attiny13.upload.using=arduino:arduinoisp
attiny13.upload.maximum_size=1024
attiny13.upload.speed=250
attiny13.bootloader.low_fuses=0x68
attiny13.bootloader.high_fuses=0xFF
attiny13.build.mcu=attiny13
attiny13.build.f_cpu=128000
attiny13.build.core=core13
```

```
#####
```

```
attiny13at4.name=ATtiny13 @ 4.8MHz (internal 4.8 MHz clock)
attiny13at4.upload.using=arduino:arduinoisp
attiny13at4.bootloader.low_fuses=0x69
attiny13at4.bootloader.high_fuses=0xff
attiny13at4.upload.maximum_size=1024
attiny13at4.upload.speed=9600
attiny13at4.build.mcu=attiny13
attiny13at4.build.f_cpu=600000
```

```
attiny13at4.build.core=core13
```

```
#####
```

```
attiny13at9.name=ATtiny13 @ 9.6MHz (internal 9.6 MHz clock)
attiny13at9.bootloader.low_fuses=0x7a
attiny13at9.bootloader.high_fuses=0xff
attiny13at9.upload.maximum_size=1024
attiny13at9.build.mcu=attiny13
attiny13at9.build.f_cpu=1200000
attiny13at9.build.core=core13
```

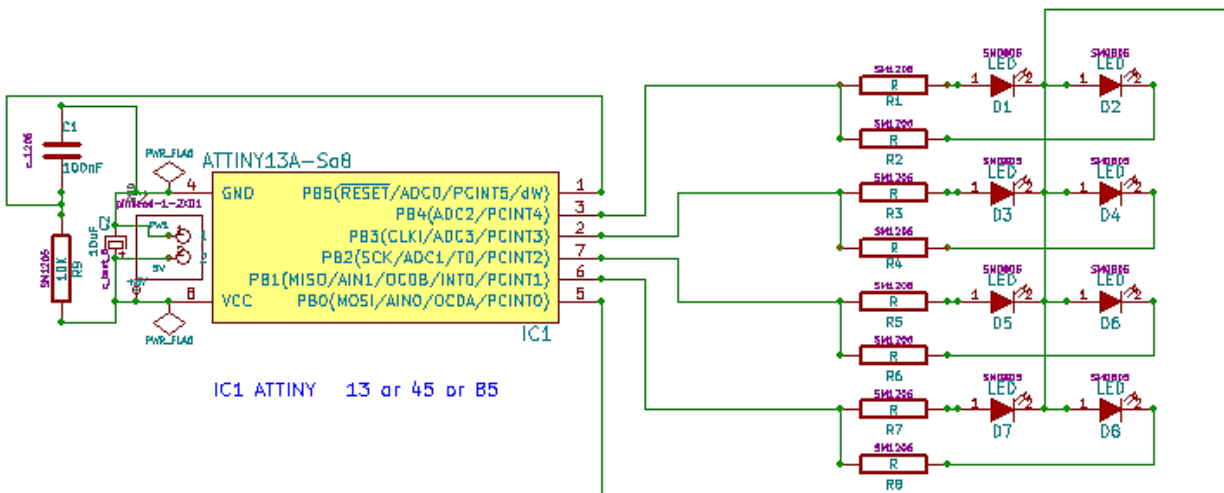
### Maintenant vous pouvez programmer l'ATtiny 13 avec Arduino:)

Lancer Arduino, vérifiez que dans le menu **outils/type de carte** vous trouvez Attiny13, si tout est ok, écrivez votre programme et compilez le pour une raison que je ne connais pas il ne faut pas dépasser 800 octets sur Attiny13 bien que la mémoire soit de 1024 octets( mon programme fait 798 octets) maintenant choisissez parmi les 3 vitesses d'horloge du menu **type de carte**, une fois la vitesse choisie, vous choisir votre programmeur pour cela allez dans le menu **outils/programmeur** (j'utilise USBasp) puis vous devez programmer les fusibles (avant de téléverser votre programme compilé), pour cela aller dans le menu **outils/graver la séquence d'initialisation** ,cela fait téléversez votre programme.

Il existe aussi des Cores pour les Attiny 45 et 85 sur ce lien par exemple :

<https://code.google.com/p/arduino-tiny/>

Voici le schéma Kicad de mon sapin :



## et voici le sketch arduino

```
/*
 * TreeOneByOne.ino For ATtiny 13
 *
 * Created: 26/07/2015 14:42:12 Arduino 1.06
 * Author: Mic-Josi
 * Attiny13
 * for Tiny Tree Rev. 1.1 and 1.7
 * pour Tiny Tree Rev. 1.1 et 1.7
 * configuration sélection oscillateur interne 4.8 Mhz et division par 8 coupure alimentation à 1.8 Volts
 * setting internal oscillator 4,8 MHz and division by 8 selected brown-out level detection at 1.8 Volts
 * FuseBits: Low = 0x69 High = 0xFD
 */

/*
These inused #define are for using with Amtel Studio 6 only
Ces défines sont à utiliser uniquement avec Amtel Studio 6
# define __DELAY_BACKWARD_COMPATIBLE__
# define F_CPU 8000000UL
# include <avr/io.h>
# include <util/delay.h>
# define byte uint8_t
*/

//Arduino defines
#include <avr/pgmspace.h>
#define LED_COUNT 9
#define DDR_BYTE 0
#define PORT_BYTE 1

const byte matrix[LED_COUNT][2] PROGMEM = {

    // DDR_BYTE      PORT_BYTE
    {0b00010001 , 0b00010000},//L0-PB4      0
    {0b00010001 , 0b00000001},//L1-PB4      "  "
    {0b00001001 , 0b00001000},//L2-PB3      " 1 2 "      ** Pin-Tree **
    {0b00001001 , 0b00000001},//L3-PB3      " 3  "      L0+ L1- Pin 3
    {0b00000101 , 0b00000100},//L4-PB2      " 4      6 "      L2+ L3- Pin 2
    {0b00000101 , 0b00000001},//L5-PB2      "      "      L4+ L5- Pin 7
    {0b00000011 , 0b00000010},//L6-PB1      " 5      7 "      L6+ L7- Pin 6
    {0b00000011 , 0b00000001},//L7-PB1      "*****"      ** +- Commun PB0 Pin 5 **
};

//*****
void turnOn( byte led ) // PBx selection
{
    DDRB = pgm_read_byte (&( matrix[led][DDR_BYTE]));
    PORTB = pgm_read_byte (&(matrix[led][PORT_BYTE]));
}
```



```

        figure [l] = 19;
        break;
    }

}
for( byte l = 0; l < LED_COUNT; l++ )
{
    allume = figure[l];
    turnOn(allume);
    delay(sp);
}

    light ++;
}
delay(50);

}

//*****

void setup()
{
}

//*****

//=====
void loop()
{
    /* 0 à 7 emplacement d'allumage de la LED,8 LED éteinte
    0 to 7 lighting location of the LED, 8 LED off */

    SomeOne (8,0,1,2,3,4,6,5,7,150); // une par une tres lent
    SomeOne (8,0,1,2,3,4,6,5,7,65); // une par une moyen
    SomeOne (15,0,1,2,3,4,6,5,7,25); // une par une rapide
    SomeOne (90,0,1,2,3,4,5,6,7,1); // flash
    SomeOne (8,0,1,2,8,8,8,8,8,70); // petit triangle haut lent
    SomeOne (8,1,2,3,8,1,2,3,8,49); // petit triangle haut inversé lent0
    SomeOne (8,5,4,1,0,0,2,6,7,42); // grand triangle lent
    SomeOne (8,8,5,4,1,3,2,6,7,46); //grand M rapide
    SomeOne (8,1,2,6,4,3,8,8,0,35); //trapèze&L4 Rapide
    SomeOne (10,0,1,2,8,4,5,6,7,40); //grandTriangleparL1
    // SomeOne (8,0,2,1,8,4,6,5,7,150); //02184657

}

//=====

// attention le programme compilé ne doit pas dépasser 875 octets sinon blocage à l'exécution

```

**Voici des vidéos sur Youtube :**

<http://youtu.be/vjNmg8G5stg>

[https://youtu.be/Hd\\_K3JuI6Yk](https://youtu.be/Hd_K3JuI6Yk)

**voici un lien intéressant pour vos idées cadeaux :**

<http://elabz.com/pimp-your-chocolates-with-arduino-ide-and-attiny13/>

