


	Classe : 1 STI2D
--	------------------

	<h1>Initiation Arduino</h1> 	1 STI2D
		SIN

Mise en situation :

On souhaite modifier les caractéristiques fonctionnelles du portail.

On utilise pour cela un outil de développement qui utilise des cartes électroniques ARDUINO.

Dans un premier temps on s'intéresse à la mise en œuvre de celles-ci par un premier TP d'initiation.

Conditions de réalisation

Documentation ARDUINO

Evaluation

La semaine suivante : **NON**

Perception personnelle du TP

Appréciation du TP (cases à cocher)	Faible	Moyen(ne)	Grand(e)
Clarté du questionnement			
Difficulté			
Longueur			
Intérêt			

Acquisition des points clés du TP

Je sais utiliser le logiciel ARDUINO		
Je sais programmer des E/S		
Je sais valider le fonctionnement de mon programme.		

Activité 1: Qu'est-ce qu'un ARDUINO ?

1- Documentation :

1-1 Lire le document arduino_presentation.pdf puis le document Arduino.pdf

Activité 2: Comment faire clignoter une LED ?

1 - Préparation :

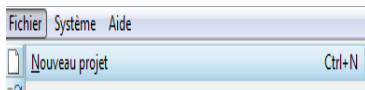
- Lire les pages 5 à 13 du document LivretArduino.pdf .

1 Création d'un projet

1.1 Saisie de schéma

Dans un premier temps nous allons créer un projet puis un schéma basique.

- Lancez Proteus 8 .
- Créez un projet par

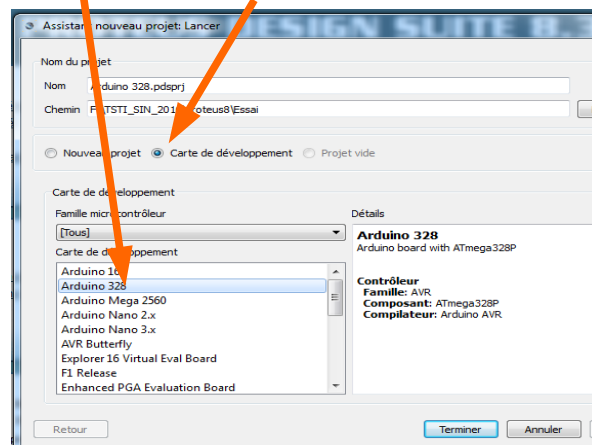


- Placez votre projet dans votre répertoire de travail situé sur le **disque dur de votre ordinateur** et **non sur le réseau ou encore dans mes documents !!!**

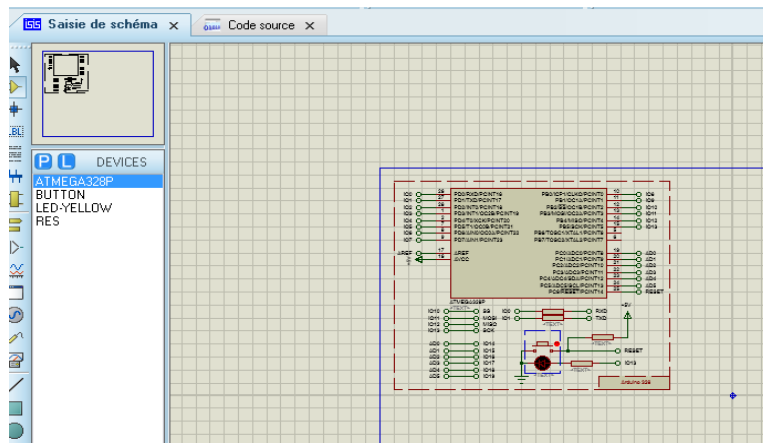
Pas d'accents et d'espaces dans vos noms de fichiers et répertoires et pas de caractères spéciaux !!!



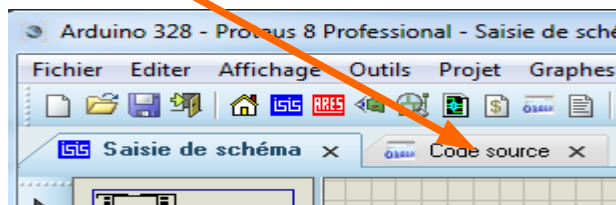
- Choisissez votre carte de développement.



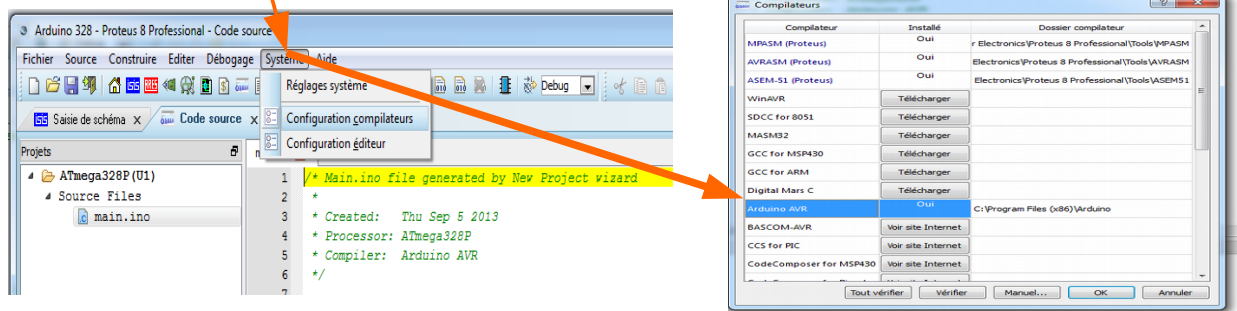
- Vous devez obtenir quelque chose ressemblant à cela !!!



- Si vous cliquez sur code source, vous obtiendrez le **code de votre arduino !**



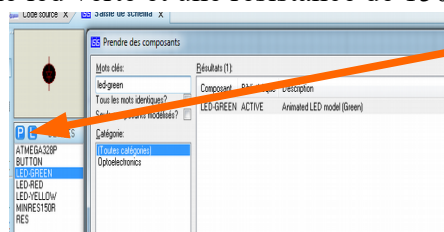
- **Vérifiez** si le compilateur pour arduino est bien installé (sinon installez le !!!)



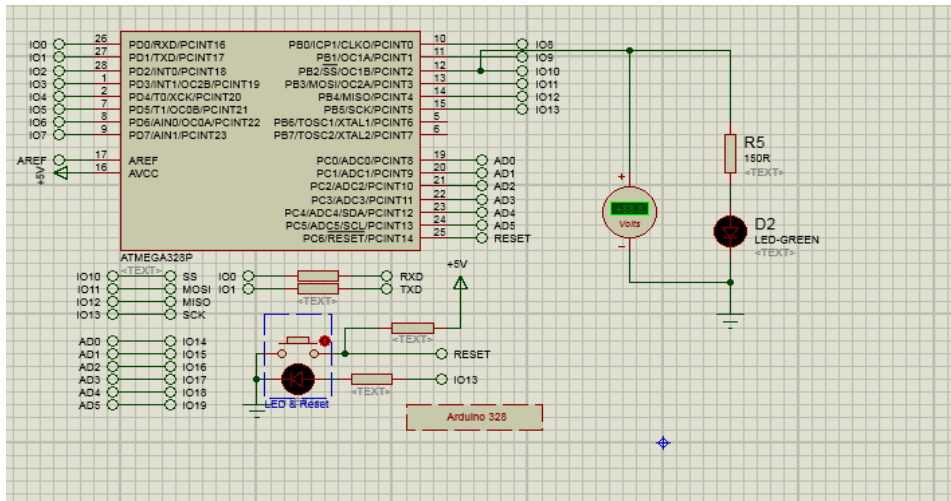
- Revenez dans votre éditeur de schéma.



- Placez une led verte et une résistance de 150 ohms avec l'outil de placement des composants



- Reliez les composants comme ci-dessous.

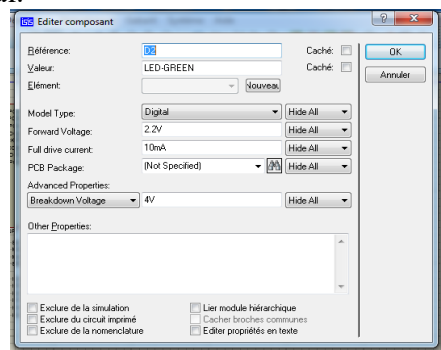


Pour vous aider, utilisez le document isis_demarrage.pdf

Les documents suivants sont encore d'actualité, même si la forme a changé.

Vous pouvez donc, consulter les documents ressources fournis.

- Cliquez avec le bouton droit de la souris sur la diode, puis dans les propriétés ; passez le « Model Type » en digital.



1.1 Édition du code source

On se propose de faire clignoter la Led !!! Pour cela il y a deux moyens possibles. Soit on récupère un code déjà écrit, soit on l'écrit soit même .

1.1.1 Utilisation de l'éditeur intégré

- Ouvrez l'éditeur de code de Proteus.



- Complétez le code ci-dessous

```
void setup()
{ //Initialisation de l'arduino
```

```
pinMode(.....); // Configuration de la broche en Sortie
```

```
}
```

```
void loop()
```

```
{ // Programme principal  
  digitalWrite(.....); // Led allumée  
  delay(...);           // attendre 1S  
  digitalWrite(....., ...); // Led éteinte  
  delay(.....);         // attendre 1S  
}
```

Vous avez à votre disposition le document ressource

[Arduino - Premiers pas en informatique embarquée.pdf](#)

http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Reference

Attention !!!!



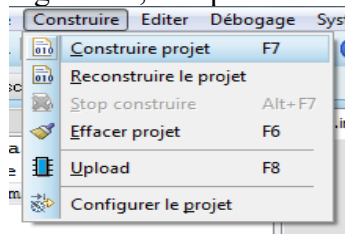
Dans les documents ressources ou sur internet, les solutions toutes faites sont disponibles etun copier-coller est tellement tentant !!!!

Meilleure façon de ne rien apprendre !!!

Prenez le temps de saisir le code et de rechercher la signification de chaque commande ainsi que d'explicitier chaque ligne de votre programme !!!

Écrivez un **algorithme** sur un papier, **préparez votre travail** et **réfléchissez avant de vous jeter sur le clavier** et vous progresserez infiniment plus vite

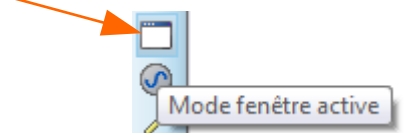
- Après avoir complété votre programme, compilez celui-ci par



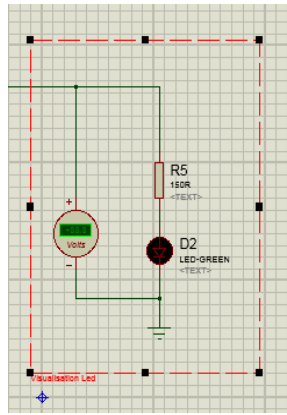
Corrigez vos éventuelles erreurs .

Pour pouvoir visualiser votre code et votre simulation sur le schéma, il faut définir les zones interactives.

Pour cela on utilise l'outil



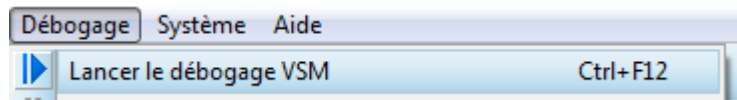
- Entourez la zone que vous souhaitez visualiser et donnez lui un nom



1.1 Le débogage

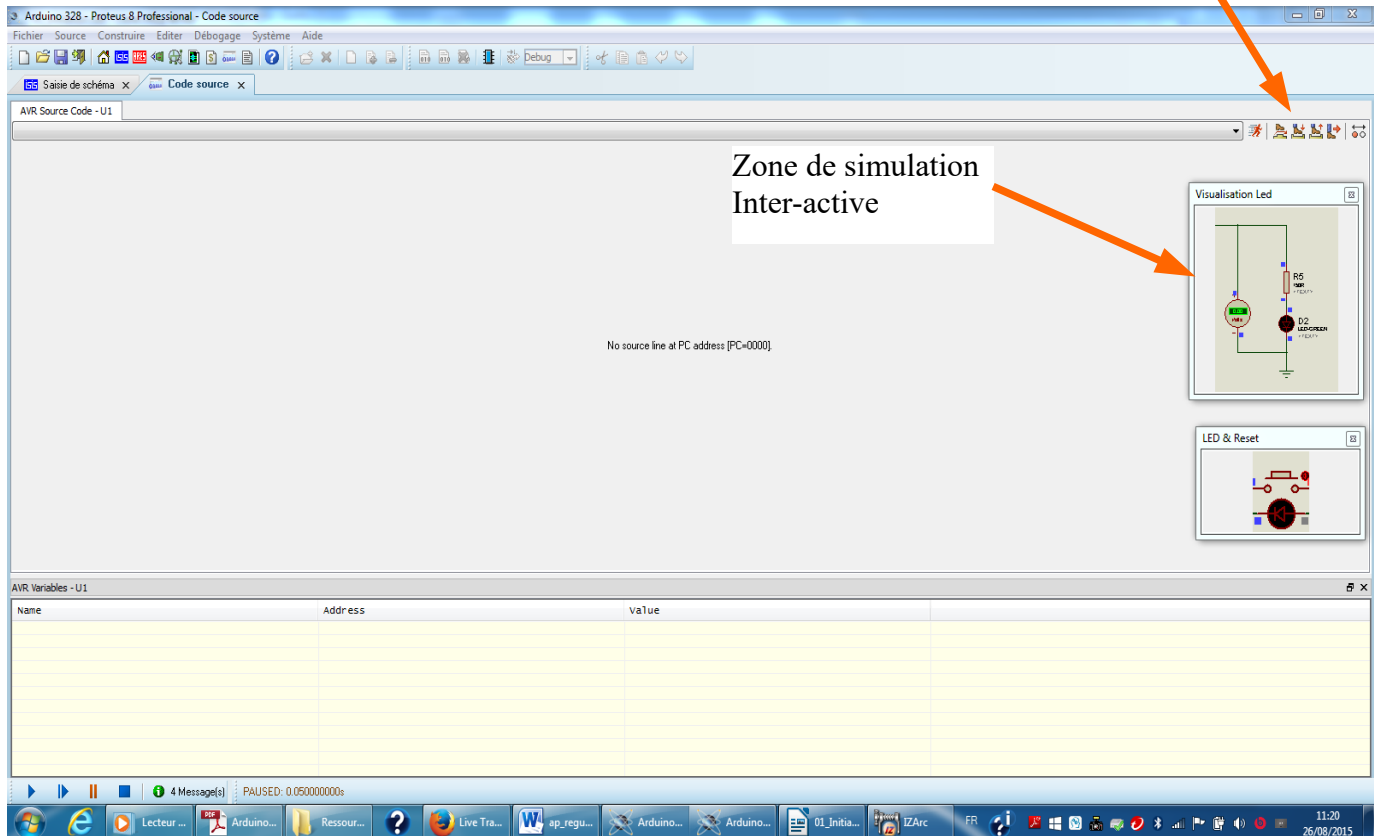
Pour pouvoir mettre au point notre programme, nous allons utiliser un « outil de mise au point » nommé debugger en anglais ou débogage en français.

- Lancez l'outil de débogage.

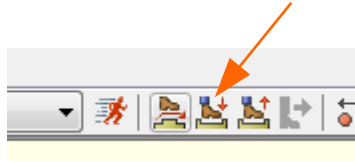


Vous devriez obtenir une fenêtre de ce style là.

Outils de débogage



- Disposez vos fenêtres de manière ergonomique.
- Cliquez plusieurs fois sur le **mode pas à pas**.



- Analysez le résultat de ces actions
- Modifiez votre programme de la manière suivante :

Utilisez deux variables, une pour la broche utilisée par la led et l'autre pour la durée de la temporisation.

dureePause=1000;

sortieLed=10;

- Insérez ces deux variables avant la fonction setup() d'initialisation.

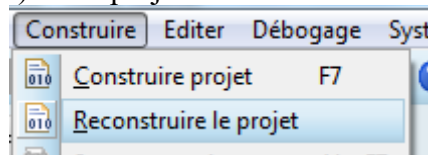
```

10 -----
11 ----- int dureePause=1000;
12 ----- int sortieLed=10;
13 -----
14 ----- void setup()
15 ----- { // put your setup code here, to run once:
16 00F6      pinMode(OUTPUT, sortieLed);
17 ----- }
18 -----

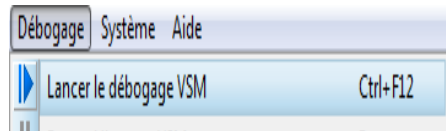
```

Ces deux variables sont des variables globales utilisables dans tout le programme.

- Recompilez (Reconstruire) votre projet.



- Relancez le débogage.



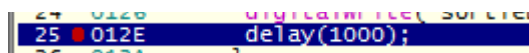
- Utilisez le mode pas à pas et analysez le contenu de la fenêtre AVR Variables

Name	Address	Type	Value
sortieLed	00800100	word	10
dureePause	00800102	word	1000

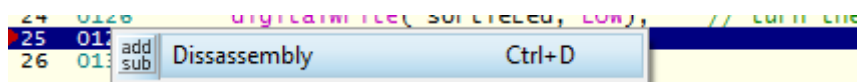
Il est possible de **poser des points d'arrêts** dans votre programme par



Cela permet de lancer le programme par jusqu'au point d'arrêt choisi dans votre programme.



Il est aussi parfois intéressant de voir le **code machine** de votre processeur, pour cela il faut utiliser le bouton droit de la souris et désassembler le code.(il faut avoir sélectionné une ligne de code auparavant)



```

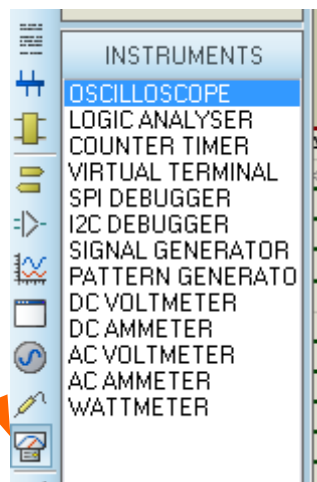
12 ----- int sortieLed=10;
13 -----
14 ----- void setup()
15 ----- { // put your setup code here, to run once:
16 00F6 pinMode(OUTPUT, sortieLed);
00F6 LDS R22,$0100
00FA LDI R24,$01
00FC CALL $0352
0100 RET
17 ----- }
18 -----
19 -----
20 ----- void loop()
21 ----- { // put your main code here, to run repeatedly:
0102 PUSH R28
0104 PUSH R29
22 0106 digitalWrite( sortieLed, HIGH); // turn the LED on (HIGH is the voltage level)
0106 LDI R28,$00
0108 LDI R29,$01
010A LDI R22,$01
010C LD R24,Y
010E CALL $03C4
23 0112 delay( dureePause); // wait for a second
0112 LDS R22,$0102
0116 LDS R23,$0103
011A CLR R24
011C SBRC R23,$07
011E COM R24
0120 MOV R25,R24
0122 CALL $023A
24 0126 digitalWrite( sortieLed, LOW); // turn the LED off by making the voltage LOW
0126 LDI R22,$00
0128 LD R24,Y
012A CALL $03C4
25 012E delay(1000);
012E LDI R22,$E8
0130 LDI R23,$03
0132 LDI R24,$00
0134 LDI R25,$00
0136 CALL $023A
26 013A }
013A POP R29
013C POP R28
013E RET

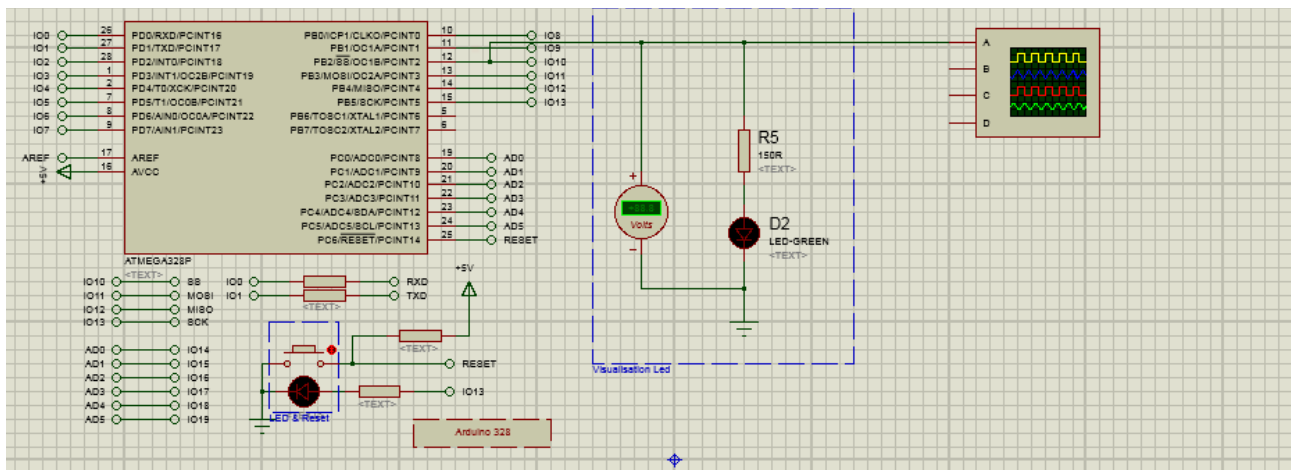
```

1.1 Placement et utilisation d'appareils de mesures

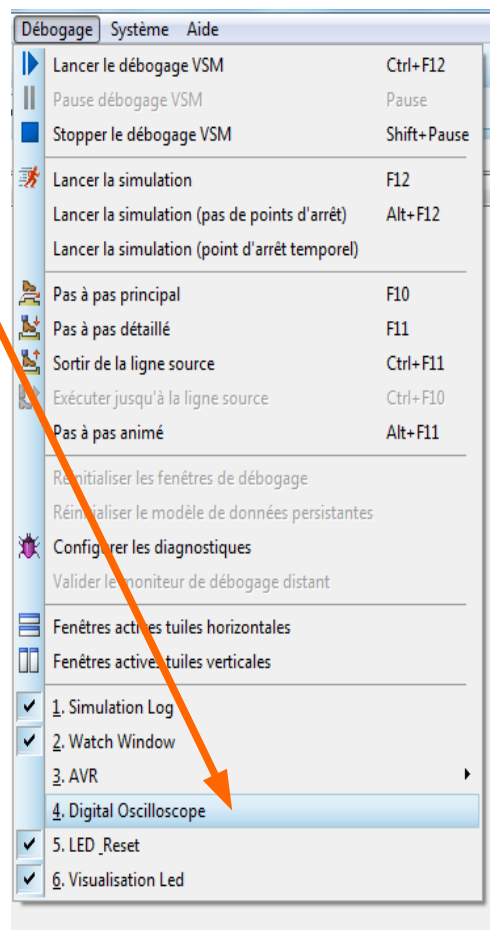
Il est possible de placer des appareils de mesures et voir l'interaction avec votre programme.

- A l'aide de **l'outil instrument**, choisissez oscilloscope puis raccordez le canal A à la sortie de votre arduino qui commande la Led.

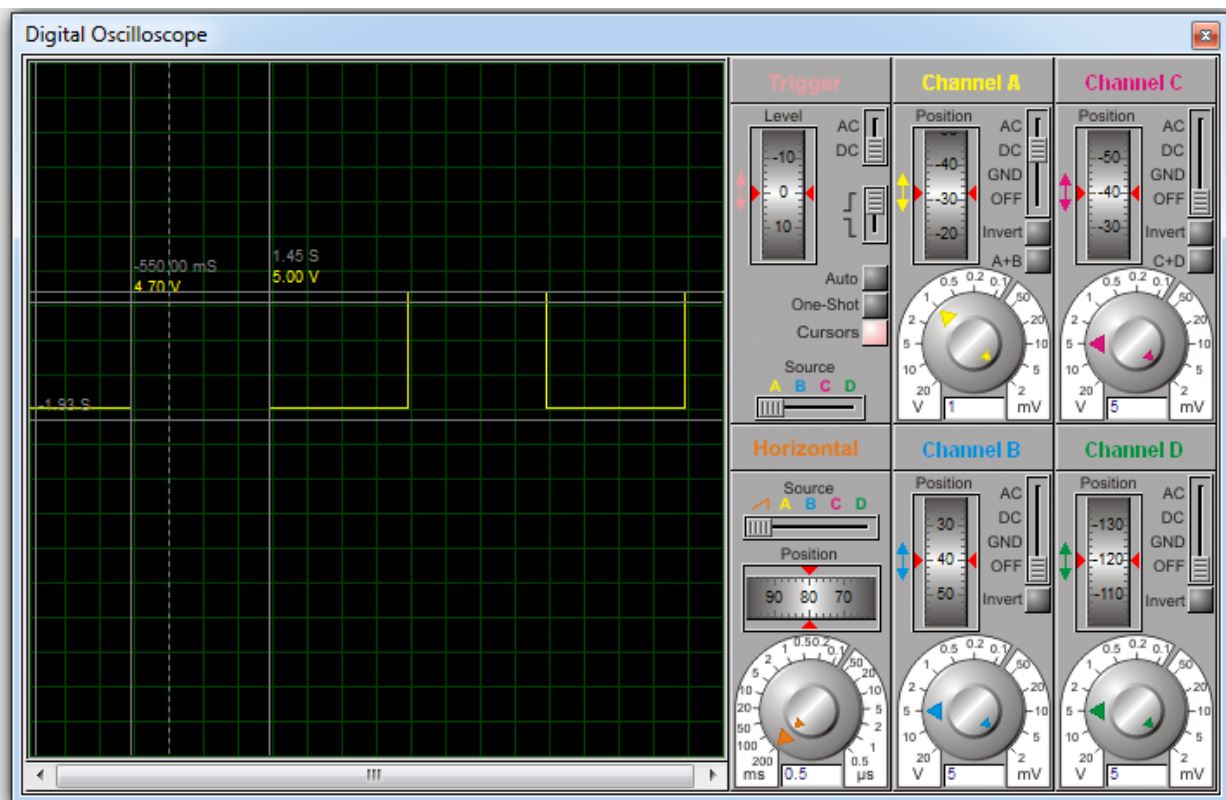




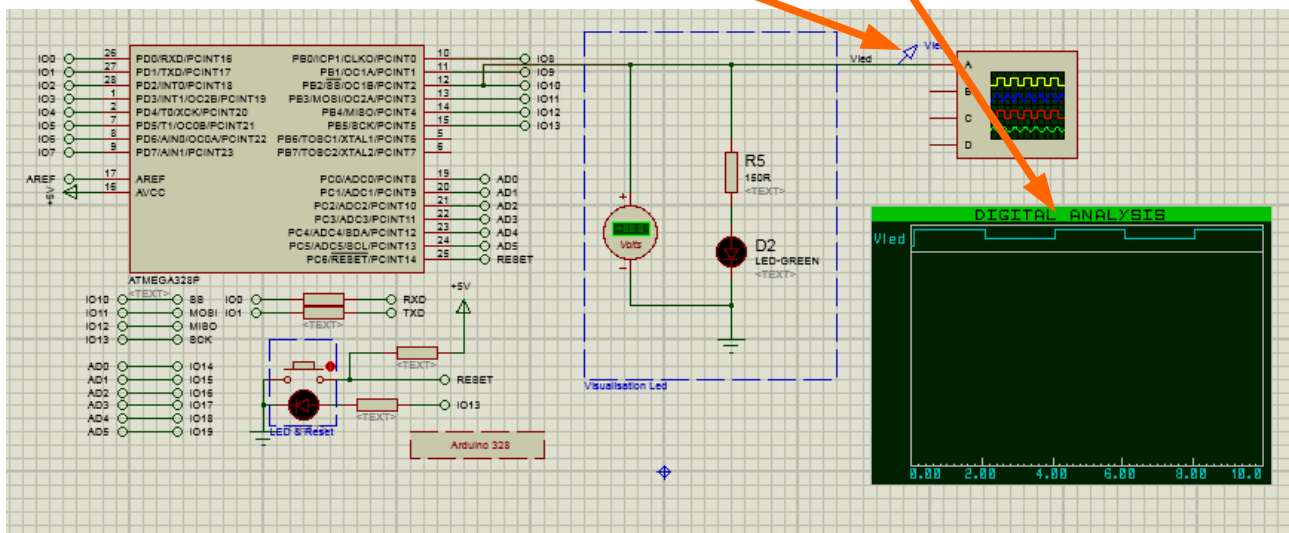
➤ Dans la fenêtre code source, lancez le débogage, puis demandez à visualiser **Digital oscilloscope**.



➤ Lancez le débogage, puis réglez l'oscilloscope afin d'obtenir un relevé correct sur celui-ci.



Vous pouvez aussi placer une **sonde de tension** et un **graphe digital** pour relever la sortie de commande de la led. (voir vos documents ressources)

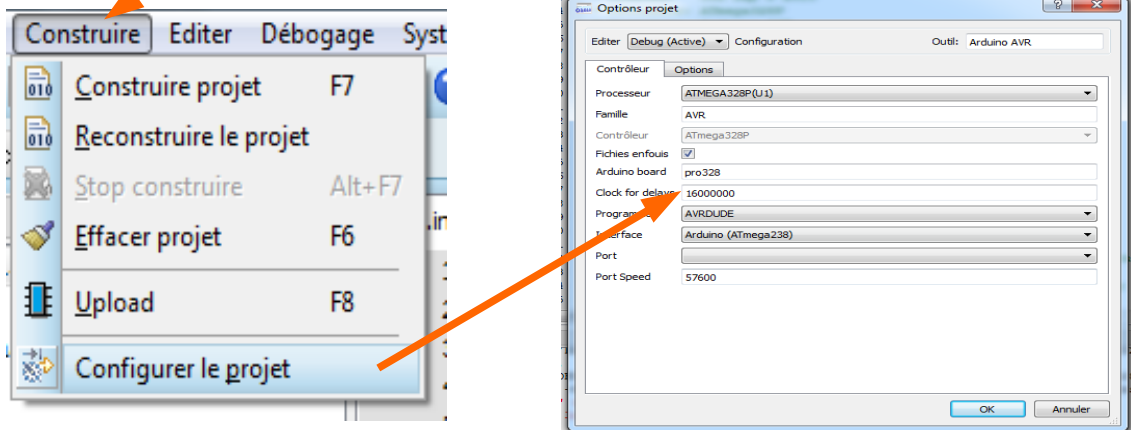


1.1 Implantation du programme

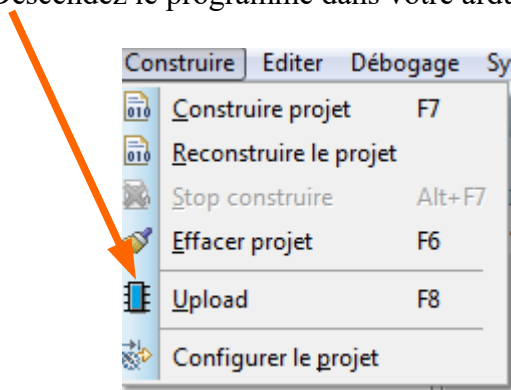
Une fois que vous avez fini de mettre au point votre programme, il est possible de venir directement programmer votre arduino.

- Connectez votre arduino.

- Configurez la fréquence de votre horloge de processeur



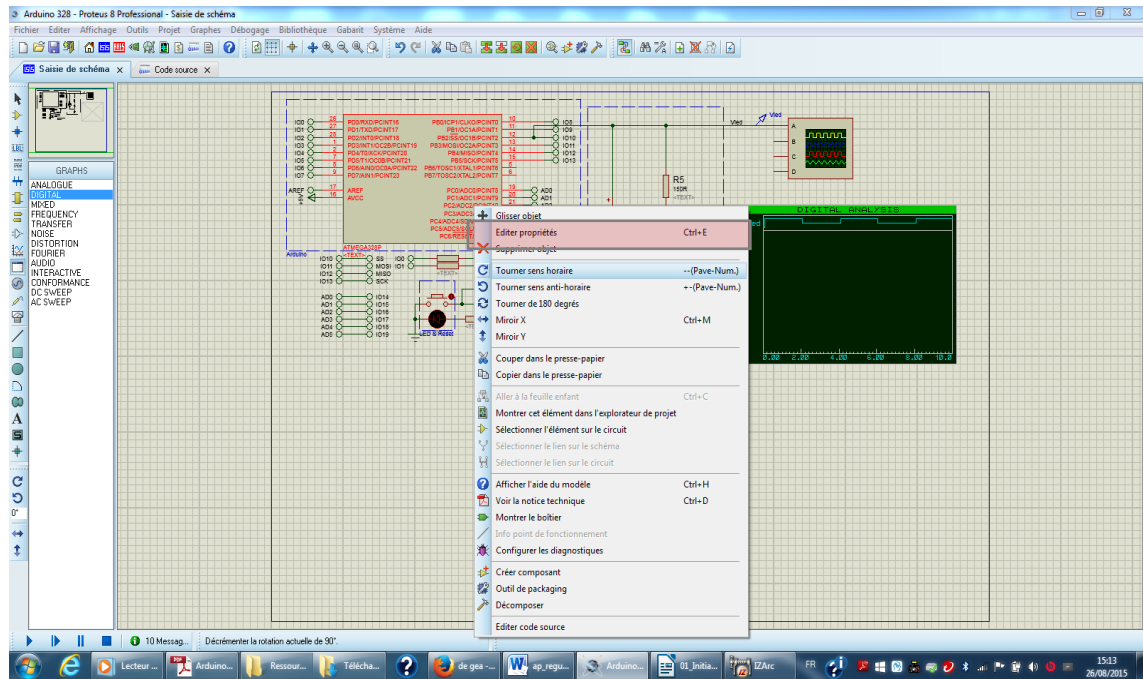
- Descendez le programme dans votre arduino .



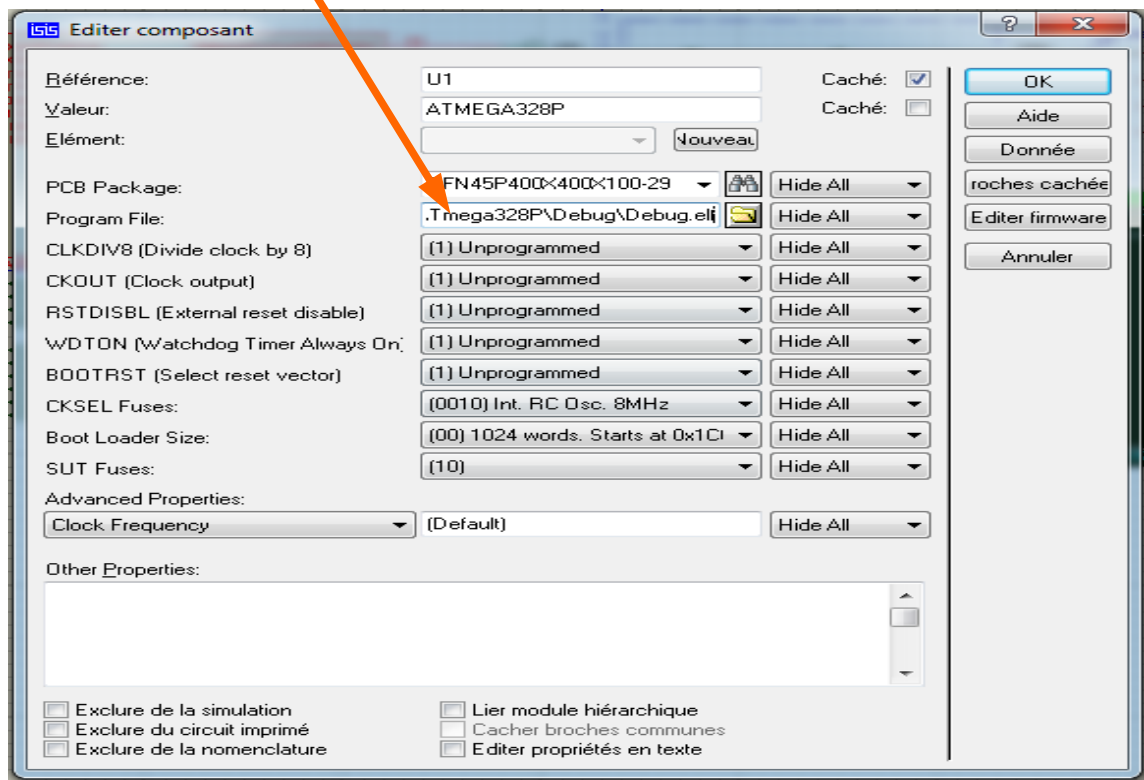
1.1 Implantation du programme dans l'outil de simulation

L'arduino utilise un fichier particulier pour simuler son fonctionnement. Ce fichier se trouve dans les propriétés de l'arduino.

- Sélectionnez l'arduino, puis bouton droit souris et propriétés.



Le fichier se trouve dans Program file



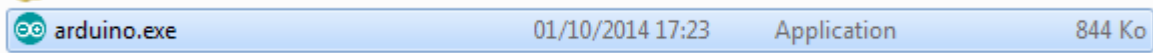
- Visualisez celui-ci et notez son extension.

Debug.elf

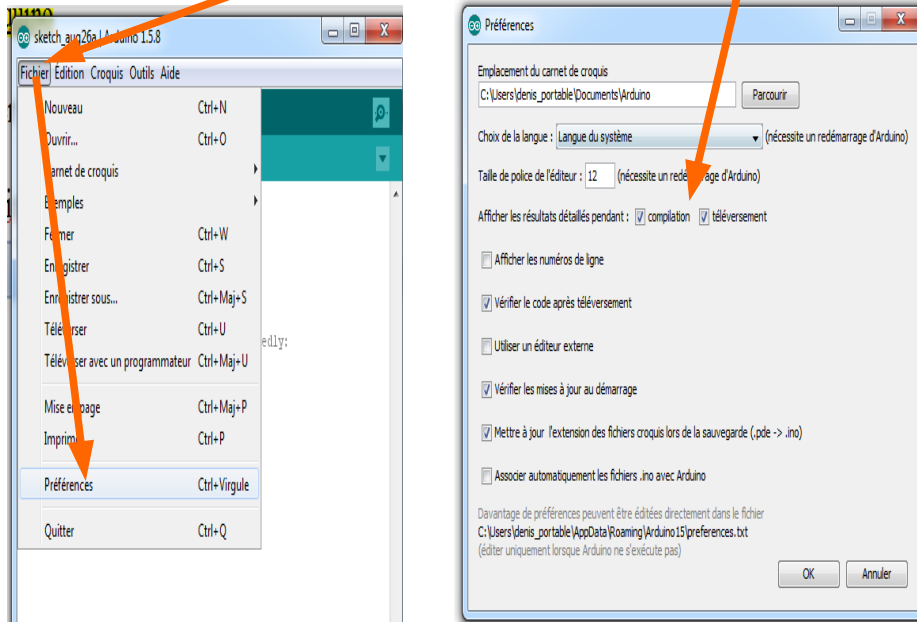
1 utilisation de l'IDE arduino

Il est possible d'utiliser les programmes développés avec l'IDE arduino

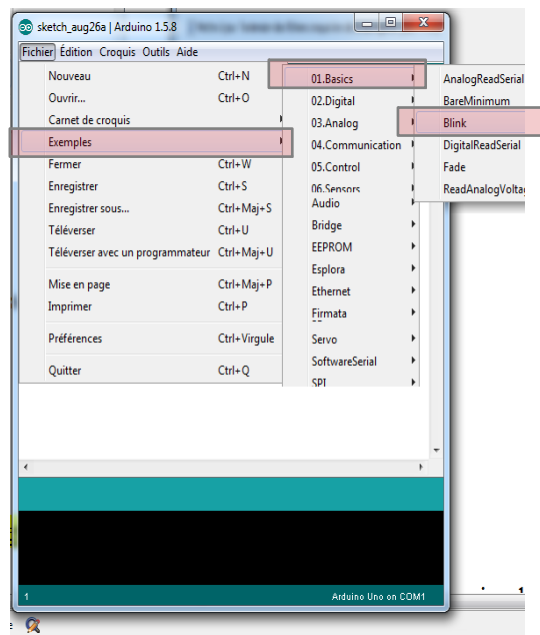
- Lancez le logiciel arduino



- Allez dans **fichier** puis **référence**, puis cochez compilation et téléversement



- Ouvrez votre IDE, allez chercher l'exemple blink



- Modifiez votre fichier pour utiliser la bonne broche de sortie de la LED.
- Compilez votre fichier

- Notez la position du fichier avec l'extension.elf.



```
Compilation terminée
C:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp\Blink.cpp.o C:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp\core.a -LC:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp -lm
C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-change-warnings --change-section-lma .eeprom=0
C:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp\Blink.cpp.elf C:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp\Blink.cpp.eep
C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy -O ihex -R .eeprom C:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp\Blink.cpp.elf
C:\Users\DENIS_-1\AppData\Local\Temp\build6576017530390422138.tmp\Blink.cpp.hex

Le croquis utilise 1 030 octets (3%) de l'espace de stockage de programmes. Le maximum est de 32 256 octets.
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2 039 octets pour les variables locales. Le maximum est de 2 048 octets.
```

- Avec votre explorateur, copiez ce fichier dans votre projet.

Exemple :

C:\Users\denis_portable\AppData\Local\Temp\build6576017530390422138.tmp

- Dans votre éditeur de schéma isis, ouvrez les propriétés de votre arduino puis donnez le chemin de ce fichier dans Program file.
- Lancez la simulation. Vous devriez avoir le même fonctionnement.

Programmer votre arduino. Pages 9 à 16.

Tester votre programme sur une carte d'essai.

Activité 3: Comment faire clignoter une LED après un appui sur un bouton poussoir ?

Le schéma de la carte interface se trouve en fin de TP

3- Programmation :

Lire les pages 16, 19,29 du document [LivretArduino.pdf](#)

Tester et mettez votre solution au point sur proteus.

Programmer votre arduino. Pages 9 à 16.

Tester votre programme sur une carte d'essai.

Activité 4: Comment faire varier l'intensité lumineuse d'une LED pendant une durée de 5 Secondes ?

4- Programmation :

Aller à la page suivante :

http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.AnalogWrite

En vous inspirant de cette documentation :

Testez et mettez au point votre solution sur proteus.

Programmer votre arduino.

Tester votre programme sur une carte d'essai.

Schéma de la carte interface :

