CCI Ile de France	PR-2101 TP introduction à Arduino Durée 8h00 Janvier 2019	E2

Remis par P. Poulichet et A. Fakri

Ce TP de 8 heures est une introduction à l'atelier PR-2101 que vous aurez ensuite. Le TP a pour but d'introduire l'utilisation de la carte arduino.

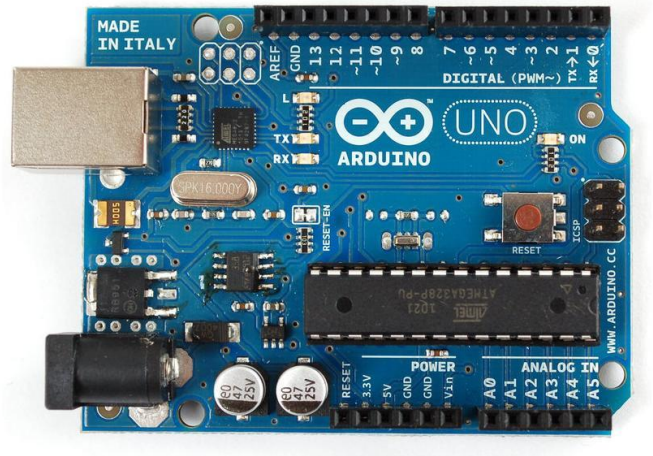
Vous devrez faire valider chaque partie réalisée. A l'issue des 8 heures, vous devrez montrer votre cahier de projets dans lequel est noté votre compte rendu de TP. Il devra y être consigné vos programmes commentés, les mesures faites, les explications du fonctionnement des capteurs… Le compte rendu est à rendre par trinôme.

Sur les PCs connectés en réseau avec le logiciel de gestion Flex, choisissez la « configuration Circuit ». Cela vous permettra d'utiliser le logiciel permettant de programmer l'arduino.

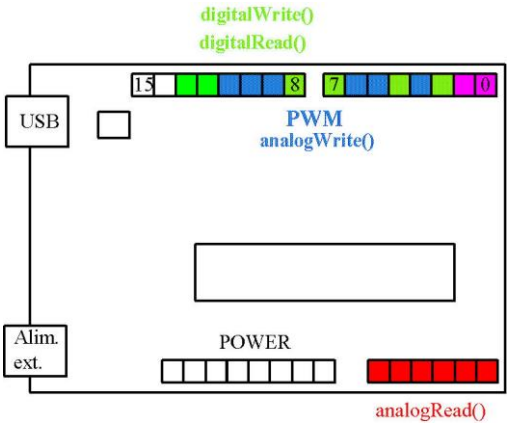
1. Installation et configuration de la carte Arduino

1.1. Carte arduino

Nous utiliserons une carte arduino uno. Elle emploi un microcontrôleur ATMEGA328P alimenté en 5 V. Il y a 14 entrées/sorties numériques dont 6 sont utilisables en PWM (Pulse Width Modulation ou MLI pour modulateur de largeur d'impulsion). Il y a 6 entrées analogiques. Le microcontrôleur possède un CAN avec 10 bits de résolution. Sur la carte, il y a un circuit qui permet de gérer facilement l'USB et qui peut alimenter la carte.



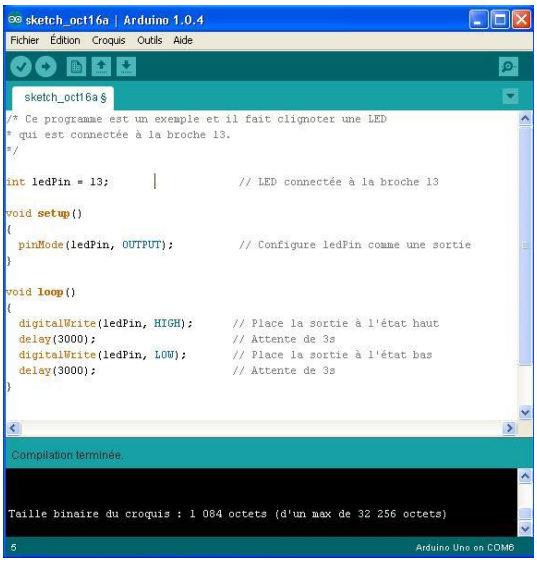
Mémoire Flash	32 ko
Mémoire RAM	2 ko
Mémoire EEPROM	1 ko
Fréquence d'horloge	16 MHz
Courant max. E/S	40 mA



Pour en savoir plus, consultez la page : http://arduino.cc/en/Main/ArduinoBoardUno

1.2. Structure d'un programme

Un exemple de programme est donné ci-dessous.



- Le commentaire de description du programme appelé sketch débute par ***/**** et se termine par ****/***.
- Puis il est placé la définition des constantes et des variables avec l'instruction *int*.
- Vient ensuite la configuration des entrées / sorties avec l'instruction ***void setup()***. La suite d'instructions est précédé de ***{*** et se termine par ***}***.
- On définit par l'instruction ***void loop()*** la programmation des interactions et comportements. La suite d'instructions est précédé de ***{*** et se termine par ***}***.

1.3. Installation de l'interface de développement

Pour utiliser l'interface de développement, télécharger le logiciel Arduino (version windows) à l'adresse : http://arduino.cc/en/Main/Software
Elle est également installée à l'ESIEE sur la configuration Circuit des PCs installés en réseau.

Ouvrir le logiciel arduino. Vérifier que la carte arduino Uno est bien prise en compte par : *Outils > Type de carte > Arduino Uno*.
Brancher la carte sur un port USB. Vérifier que le port COM est bien configuré. *Outils > Port série > COM X*.
Vérifier que le port COM X est bien reconnu par windows : *Panneau de Configuration > Système et sécurité > Gestionnaire de périphériques > Ports : Arduino Uno (COM X).*

2. Clignotement d'une LED

2.1. Modification du programme et mesure

Ouvrir le fichier Blink par : *Fichier > Exemples > 01.Basics > Blink*

Pour comprendre la syntaxe et le rôle d'une instruction, utiliser la page :
http://arduino.cc/en/Reference/HomePage

Modifier le programme de telle façon que la sortie sur laquelle on connectera la LED soit la sortie 11. Modifier les instructions pour que la période soit de 1 s et la durée d'allumage de la LED 100 ms.
Lancer l'exécution du programme en cliquant sur le bouton en dessous Edition :

Connecter une sonde de tension pour mesurer à l'oscilloscope les potentiels sur la sortie 11. Relever cette tension.
Validation

2.2. Connection des LEDs

Connecter une LED et une résistance en série sur une plaquette d'essais. Déterminer la valeur de la résistance pour avoir un courant de 10 mA.

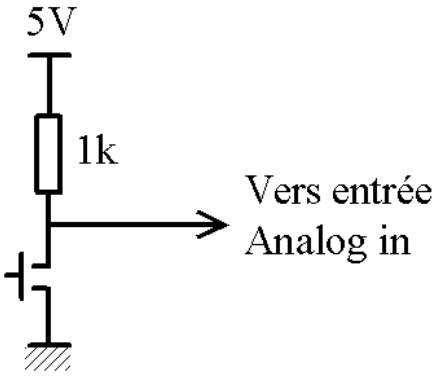
Modifier le programme pour que deux LED s'allument en opposition de phase comme dans le tableau suivant :

LED 1	Allumée	Eteinte
LED 2	Eteinte	Allumée

Validation

3. Utilisation d'un interrupteur switch

Le switch sera remplacé par un fil qui sera connecté en série avec une résistance au + 5 V comme sur le montage suivant :



Lorsque l'entrée Analog In sera au niveau 0, modifier le programme pour que la LED soit éteinte et réciproquement. On utilisera les instructions *digitalRead* et *digitalWrite*. Pour simuler un anti rebond, on pourra considérer la variation de tension aux bornes de l'interrupteur après 100 ms en utilisant l'instruction *delay().*
Validation

4. Utilisation de l'entrée analogique

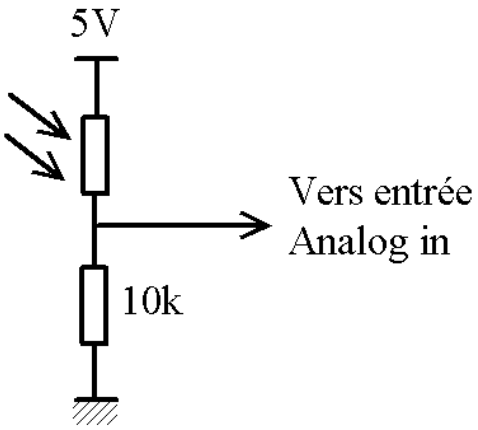
4.1. Variation de la fréquence d'allumage d'une LED en fonction de la tension d'entrée

On placera un potentiomètre de 10 k entre le +5V et la masse et le curseur sera connecté sur l'entrée Analog IN A0. On utilisera l'instruction YY = *analogRead (XX),* XX étant la valeur lue sur l'entrée Analog IN A0. L'instruction delay (YY) permettra de faire varier la fréquence de commande de la LED.

4.2. Mesure de l'éclairement

Le capteur de luminosité utilisé sera le LDR VT935G :
http://radiospares-fr.rs-online.com/web/p/ldr/4558036/
Comprendre le fonctionnement du capteur à partir de la page :
http://learn.adafruit.com/system/assets/assets/000/010/129/original/APP_PhotocellIntroduction.pdf

Quel est le principe de fonctionnement du capteur ? Télécharger la documentation du capteur. Quelle est la tension maximale d'utilisation ? Que signifie la sensibilité ? Proposez un montage permettant de détecter le passage d'un objet.



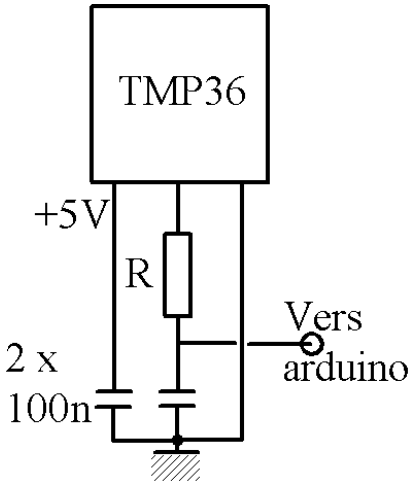
4.3. Mesure de la température

Le capteur de température utilisé sera le TMP 36. http://radiospares-fr.rs-online.com/web/p/capteurs-de-temperature-et-humidite/0427351/

Comprendre le fonctionnement du capteur à partir de la page :
http://learn.adafruit.com/tmp36-temperature-sensor/

Sur quel principe repose le fonctionnement du capteur ? Télécharger la documentation du capteur. Quelle est la tension de sortie pour une température nulle ? Quelle est la sensibilité du capteur ?
Alimenter le capteur à partir de la carte arduino. Connecter sa sortie sur une entrée analogique. Afficher sa valeur sur le port série. Pour voir la valeur, ouvrir le *Moniteur Série* par *Outils* > *Moniteur Série*. Vérifier la compatibilité de la valeur numérique lue avec la température et la valeur de la tension continue en sortie du capteur.
La tension de sortie est-elle stable ?

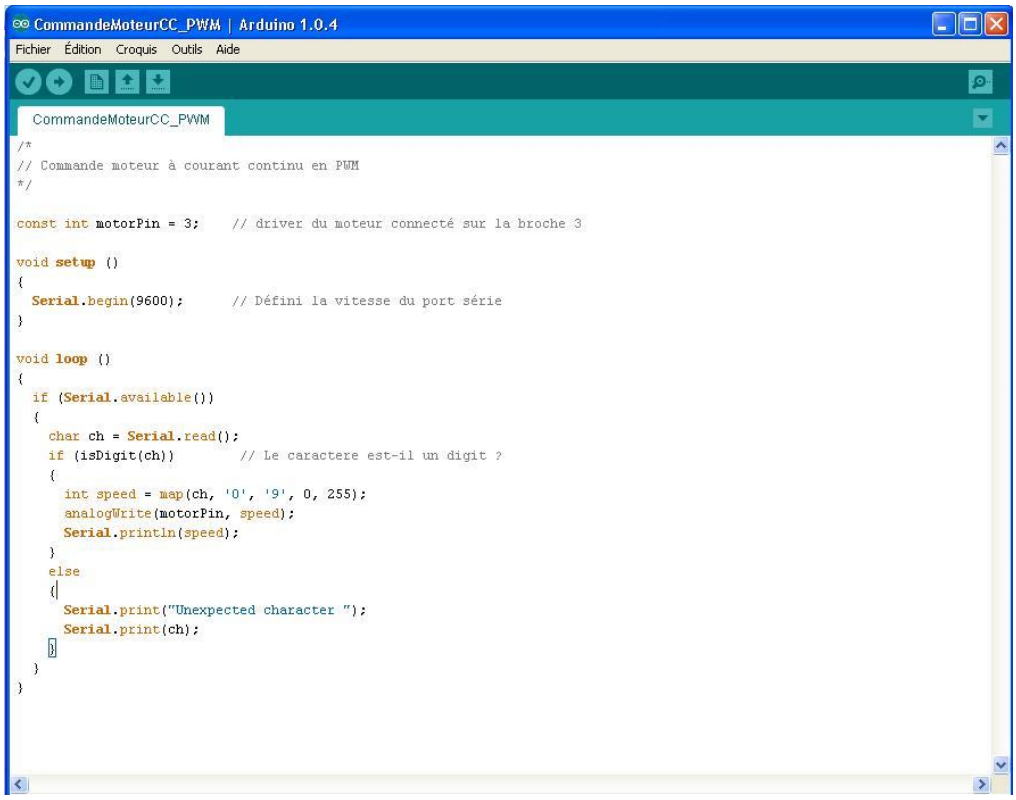
Ajouter les composants du schéma de la figure ci-dessous. Déterminer la valeur de R pour avoir une fréquence de coupure de l'ordre de 10 Hz. Quel est le rôle des composants ajoutés ?
La sortie est-elle plus stable ?
On peut également calculer la moyenne de plusieurs acquisitions successives : faire 3 acquisitions séparées de 100 ms (avec l'instruction *delay*) puis calculer sa somme divisée par 3. Ainsi, on obtient : $Vs = \frac{1}{3}\left(Acq1(0) + Acq2(100) + Acq3(200)\right)$.



Validation

5. Commande d'un moteur à courant continu en PWM

Le programme suivant est utilisé pour commander par une variation du rapport cyclique (PWM pour Pulse Width Modulation ou MLI pour Modulation de Largeur d'Impulsions) la vitesse d'un moteur en courant continu. La vitesse est entrée au clavier par un caractère ch compris entre 0 et 9 qui est alors converti en valeur de 0 à 255 avec l'instruction *map*. Pour voir la valeur, ouvrir le Moniteur Série.



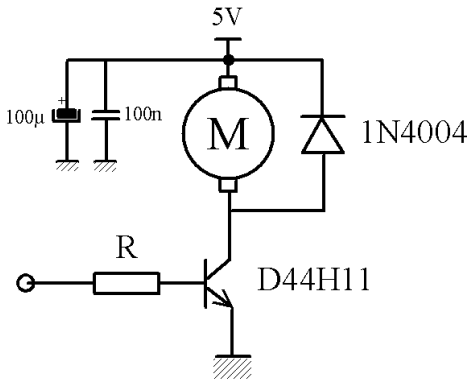

```
/*
// Commande moteur à courant continu en PWM
*/

const int motorPin = 3;    // driver du moteur connecté sur la broche 3

void setup ()
{
  Serial.begin(9600);      // Défini la vitesse du port série
}

void loop ()
{
  if (Serial.available())
  {
    char ch = Serial.read();
    if (isDigit(ch))        // Le caractere est-il un digit ?
    {
      int speed = map(ch, '0', '9', 0, 255);
      analogWrite(motorPin, speed);
      Serial.println(speed);
    }
    else
    {
      Serial.print("Unexpected character ");
      Serial.print(ch);
    }
  }
}
```

On connectera la sortie arduino utilisée au montage suivant :



Une alimentation extérieure délivre la tension 5V nécessaire pour alimenter le moteur. Attention au sens de branchement de la capacité de 100 µF car elle est polarisée. Les masses de l'alimentation et de l'arduino doivent être reliées. La documentation du transistor est disponible à l'adresse :
http://radiospares-fr.rs-online.com/web/p/transistors-bipolaires/6255076/

Déterminer la valeur de la résistance R à partir du gain en courant β du transistor et du courant circulant dans le moteur qu'on estime à 500 mA. Quel est le rôle de la diode et des capacités de découplages ? Mesurer la tension délivrée par l'arduino et la tension VCE du transistor.

Validation

6. Affichage d'une grandeur sur un afficheur LCD

Nous utiliserons l'afficheur décrit sur cette page : http://www.adafruit.com/products/715.

Puis cliquer en bas de la description > Product page with tutorials, documentation and assembly information. Puis *Using the RGB Shield* pour arriver à la page http://learn.adafruit.com/rgb-lcd-shield/using-the-rgb-lcd-shield. La « library » téléchargée doit s'installer dans le répertoire : *C:\Program Files\arduino\libraries*

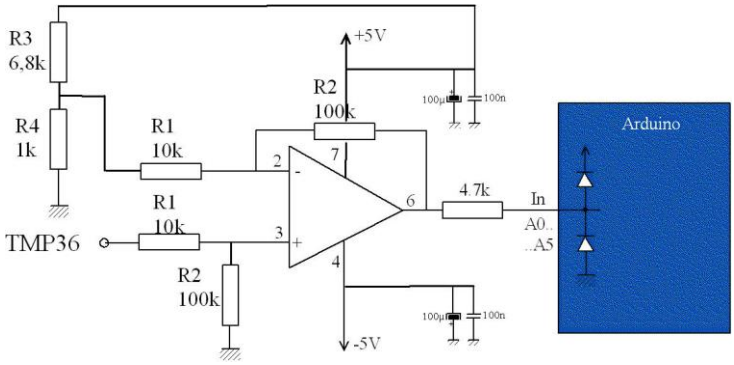
Vous devez alors retrouver dans arduino la librairie :
Fichiers > Exemples > adafruit_RGB__LCD_Shield_Library_Masters > HelloWorld.
Modifier le fichier et enregistrer le dans un croquis.

Connecter le capteur de température à une entrée analogique de l'arduino. Afficher le code numérique sur l'afficheur.
Pour convertir la grandeur affichée en degré, on tiendra compte du fait que le convertisseur analogique numérique interne délivre un mot de 10 bits pour une tension variant de 0 à 5 V en entrée. Consulter la documentation du capteur TMP36 pour afficher la valeur en degré sur l'afficheur. Ajouter un filtre lisseur « moyenneur » comme au paragraphe 4.3.

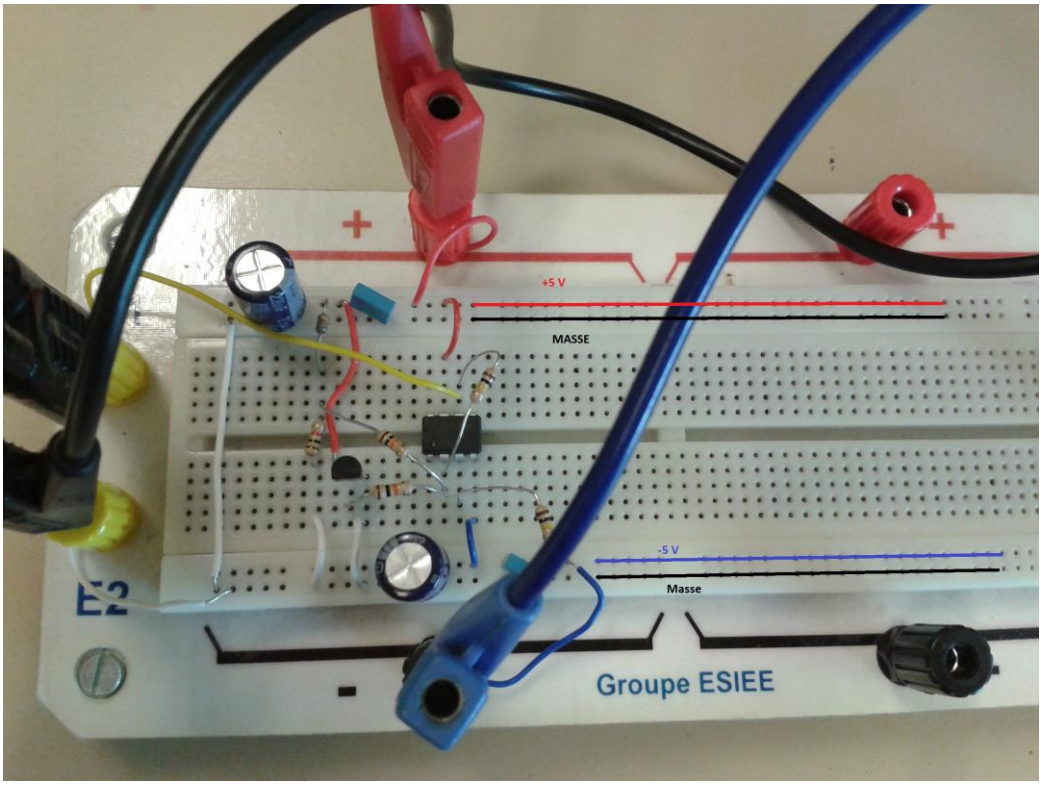
Validation

7. Utilisation d'un AOP

Le montage suivant est destiné à amplifier la tension délivrée par le capteur de température. La tension continue de sortie de l'AOP dépend de la différence des deux tensions délivrées par le pont de résistances R3-R4 et par le capteur de température TMP 36.



Sur la photo suivante, il vous est montré la façon de procéder pour câbler le montage. Vous devez utiliser une ligne pour distribuer l'alimentation +5 V et le -5 V, et deux lignes pour la masse. Cela permet de placer les capacités de découplage d'alimentation pour qu'elles soient efficaces pour réduire le bruit sur l'alimentation.



Etablir l'équation donnant la tension en sortie de l'AOP en fonction des éléments du montage et de la tension délivrée par le capteur de température.

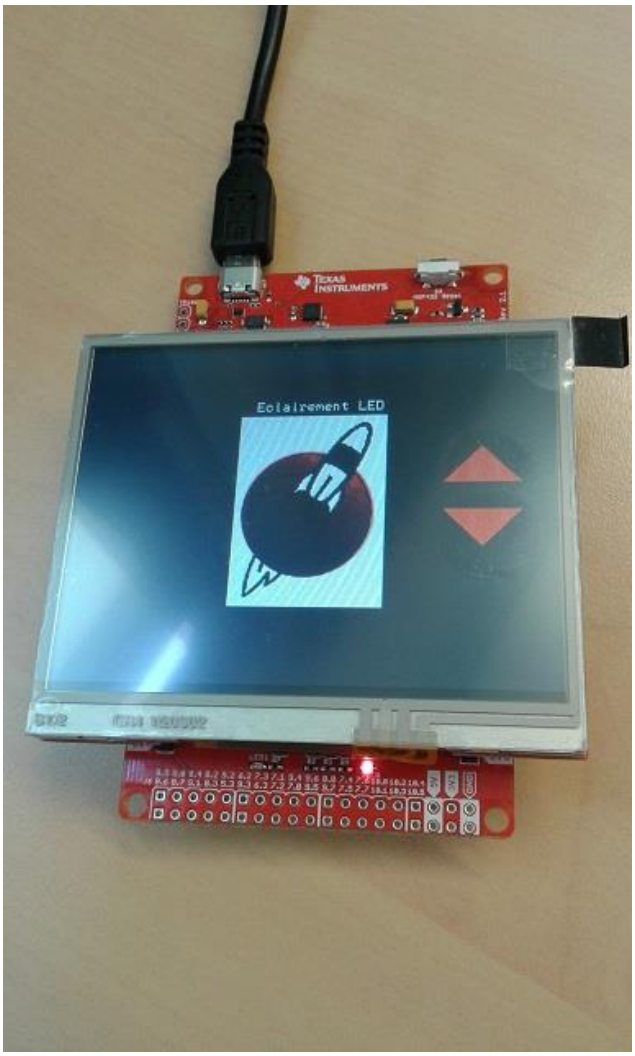
Modifier le code de l'arduino pour que l'afficheur affiche la température en degré Celsius.

Validation

Le paragraphe suivant est à tester si vous voulez utiliser lors du projet un microcontrôleur plus performant et avec des possibilités graphiques bien plus importantes que l'arduino.

8. Utilisation du microcontrôleur MSP432 et de l'écran BOOSTXL-K350QVG-S1

Le MSP432 utilise un cœur Cortex M4-F, une horloge à 48MHz, 256kB de mémoire Flash, 64kB de mémoire SRAM, 32kB de mémoire ROM, 16 timers, 2 timers 32 bits, une liaison I2C, SPI, UART, un ADC sur 14 bits 1Ms…
http://www.ti.com/tool/MSP-EXP432P401R#1



La page suivante initie à l'utilisation du MSP432 associé à l'afficheur tactile et graphique BOOSTXL-K350QVG-S1. Cet afficheur possède une librairie graphique pour afficher des cercles, des rectangles, des triangles et détecter l'appui sur l'écran…
https://perso.esiee.fr/~poulichp/Energia/Energia.html

9. Références

[1] Arduino Cookbook. Recipes to Begin, Expand, and Enhance Your Projects. By Michael Margolis
[2] Arduino - Maîtrisez sa programmation et ses cartes d'interface (shields). Christian Tavernier. Dunod.
[3] Arduino applications avancées. Christian Tavernier. Dunod.

[4]
http://air.imag.fr/mediawiki/index.php/Travaux_Pratiques_Arduino#Travaux_Pratiques_:_Faire_parler_les_choses_avec_Arduino
[5] Liste et syntaxe des instructions : http://arduino.cc/en/Reference/HomePage