PROGRAMMATION DES MICROCONTRÔLEURS

TP ARDUINO TICK TOCK

OBJECTIFS

- Lecture d'une grandeur analogique
- Programmation en C Arduino : lecture et écriture d'entrées/sorties logiques (**TOR** : **T**out **O**u **R**ien) et structures de contrôles

PRÉALABLE:

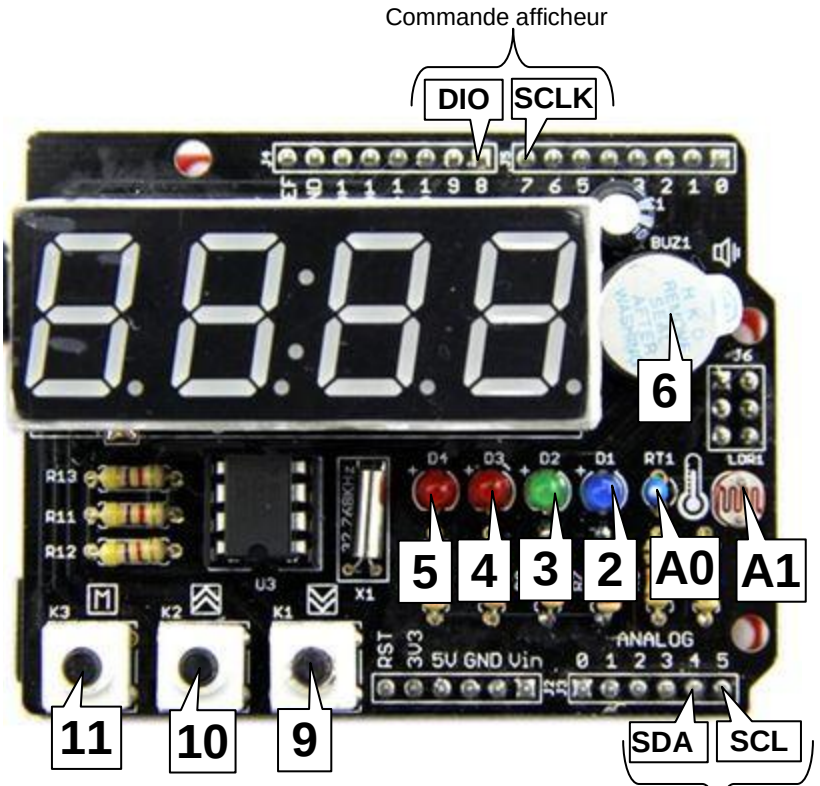
- Le tick tock shield doit être enfiché sur la carte arduino UNO, elle-même reliée au PC par une liaison USB. Celle-ci alimentera la carte UNO et son shield et permettra le téléversement du fichier exécutable relatif à votre programme.
- Coté PC, créer un dossier « TP Tick Tock 1 » qui contiendra les programmes Arduino (croquis).

PRÉSENTATION DE LA CARTE TICK TOCK SHIELD

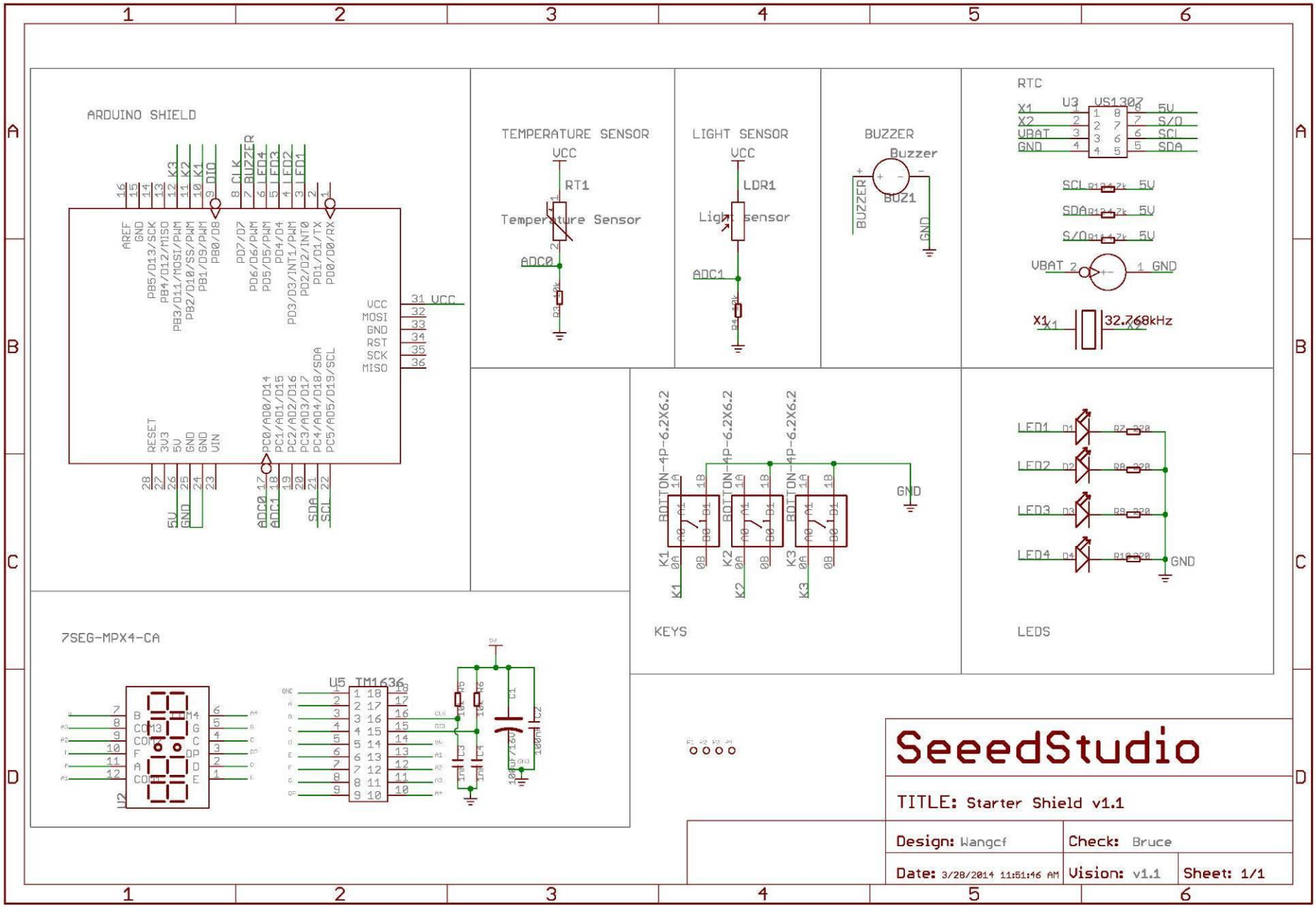
Cette carte est spécialement conçue pour l'initiation et la découverte de la programmation sur Arduino. Elle est prévue pour être enfichée sur une carte Arduino.

La platine dispose :

- d'un afficheur 7 segments 4 digits multiplexés avec son circuit de commande
- d'une horloge temps réel à quartz avec sa pile pour conserver les données horaires
- d'un capteur de température de type thermistance
- d'un capteur de lumière de type photorésistance (LDR)
- d'un buzzer
- de 3 boutons poussoirs
- de 4 leds



Les numéros correspondent aux numéros de broche de la carte Arduino


ARDUINO SHIELD
TEMPERATURE SENSOR
LIGHT SENSOR
BUZZER
RTC
KEYS
LEDS
7SEG-MPX4-CA
SeeedStudio
TITLE: Starter Shield v1.1
Design: Wangcf
Check: Bruce
Date: 3/28/2014 11:51:46 AM
Vision: v1.1
Sheet: 1/1

1) TEST DES POUSSOIRS, LEDS et BUZZER

Vous allez développer un programme appelé « test1 » permettant de tester les leds, les poussoirs et le buzzer ainsi que leurs liaisons à la carte Arduino UNO.

Conventions : NL0 ➔ ≅ 0 V – NL1 ➔ ≅ 5 V

Questions préalables :
D'après le schéma structurel de la carte pour quel niveau logique (NL1 ou NL0) sur les sorties de la carte UNO chaque diode s'allume-t-elle ? .....

D'après le schéma structurel de la carte, quel niveau logique (NL1 ou NL0) génère chaque poussoir quand il est actionné ? .....

Information à savoir
Quand un poussoir est relâché, on retrouve la patte d'entrée du microcontrôleur en l'air, donc sans niveau logique défini. Pour que cette entrée se retrouve au niveau logique 1, on peut par programme mettre en route une résistance de pull-up (tirage au Vcc) interne au microcontrôleur. A cet effet on utilisera dans le `setup()` l'instruction suivante :

```
pinMode(X, INPUT_PULLUP) ;    // X étant le n° de la broche UNO concernée
```

Description du programme de test « test1 »
Le poussoir K1 allumera les 4 leds et le poussoir K2 les éteindra.
Le poussoir K3 mettra en route ou non le buzzer suivant qu'il est appuyé ou relâché.

Méthode
Penser à utiliser les #define et choisir des noms évocateurs comme D1, K, etc, pour nommer les entrées/sorties.

Exemple :
Avant le setup() :

```
#define K1 9 // K1 est équivalent à 9 désormais (constante symbolique)
```

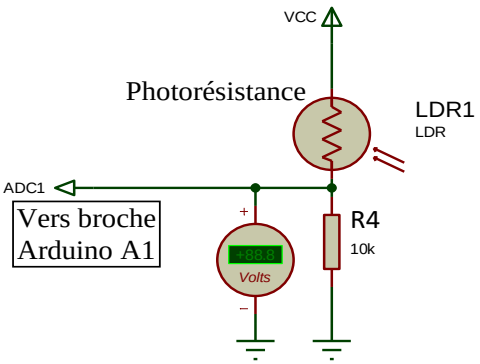
Dans le setup() :

```
pinMode(K1, INPUT);
```

- Compiler pour traduire votre code source Arduino d'extension « .ino » en un fichier binaire d'extension « .hex » contenant le code exécutable par le microcontrôleur de l'Arduino.
- Téléverser le fichier exécutable.
- Valider expérimentalement le bon fonctionnement. *Visa du prof*

2) TEST DU CAPTEUR ANALOGIQUES DE LUMIERE

Il s'agit de tester le capteur de type photorésistance dit **LDR** (**L**ight **D**epending **R**esistor) et de valider l'acquisition par la carte Arduino du signal analogique fourni ADC1.
Ci-contre on a extrait du schéma structurel complet de la carte Tick Tock la partie correspondant à la photorésistance.



A) TESTS ÉLECTRIQUES PRÉALABLES

À l'aide d'une pointe de touche et d'un voltmètre :

- Relever la tension V_{ADC1} à luminosité ambiante :
  V_{ADC1} = ……..

- En éclairant la photorésistance, vérifiez que la tension V_{ADC1} varie.
- Relever la nouvelle valeur : ……
- La tension augmente-t-elle ou diminue-t-elle ? ……
- En déduire si la résistance de la LDR augmente ou diminue avec l'éclairement ? Justifier.

……
……
……

B) ÉLABORATION DE PROGRAMMES

b.1) Programme « test2A »

Il s'agit de développer un programme « test2A » qui allume une led bleu quand l'éclairement de la LDR dépasse l'éclairement ambiant initial.

V_{ADC1} étant une grandeur analogique, on mettra en œuvre le convertisseur analogique-numérique du microcontrôleur.
Pour cela l'EDI Arduino possède la fonction **`analogRead()`** (voir description sur le site de référence du langage Arduino :
Mon-club-elec référence Arduino en français

Calculer les valeurs numériques renvoyées par **`analogRead()`** pour les deux tensions précédemment mesurées :
……
……
……
……

- Proposer un programme en C Arduino. Penser à utiliser les #define et choisir des noms évocateurs pour les entrées/sorties.
- Compiler.
- Téléverser le fichier exécutable.
- Vérifier expérimentalement le bon fonctionnement.

Visa du prof ☐

b.2) Programme « test2B »

On améliorera le programme précédent de la façon suivante : un poussoir permettra de mémoriser la luminosité ambiante, ensuite on aura le même fonctionnement que pour le programme « test2A ».

3) TEST DU CAPTEUR ANALOGIQUE DE TEMPÉRATURE

Il s'agit d'envoyer sur un moniteur via une liaison série RS232 un nombre entier image de la tension fournie par le capteur de température.

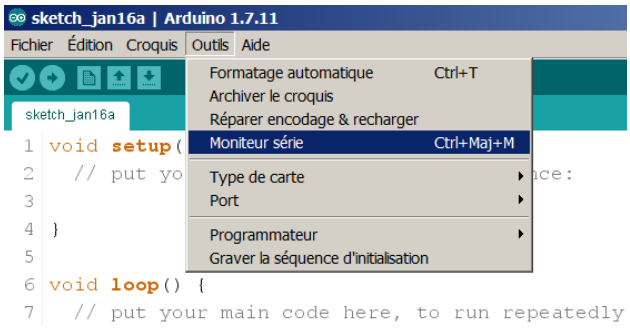
A) PRÉALABLE : UTILISATION DE LA LIAISON SERIE ASYNCHRONE

On donne le programme suivant :

SerialLink

```
1  int valeur = 0;
2
3  void setup() {
4    // put your setup code here, to run once:
5    Serial.begin(9600);
6  }
7
8  void loop() {
9    // put your main code here, to run repeatedly:
10   Serial.print("Valeur = ");
11   Serial.println(valeur);
12   valeur++;
13   delay(1000);
14 }
```

Noter que **`Serial`** est un objet inclus dans la bibliothèque par défaut et que **`Serial.begin()`**, **`Serial.print()`** et **`Serial.println()`** sont des fonctions associées à cet objet.
Les signaux de la liaison série asynchrone sont disponibles sur les broches TX et RX du microcontrôleur ATmega328P. Comme cette ligne est utilisée dans un sens (PC vers carte Uno) par l'EDI pour téléverser le programme, elle peut être utilisée dans l'autre sens (Uno vers PC) pour affiché du texte dans le moniteur série intégré à l'EDI. On s'affranchira ainsi d'un moniteur externe.



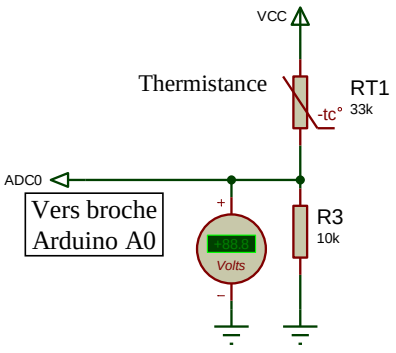
Téléverser le programme et observer son fonctionnement.

Expliquer la différence entre `print("valeur = ")` et `print(valeur)` et entre `print` et `println` ?

.....
.....
.....
.....
.....

B) TESTS ÉLECTRIQUES PRÉALABLES SUR LE CAPTEUR DE TEMPERATURE

Il s'agit de tester le capteur de température constitué d'une thermistance RT1 et de valider l'acquisition par la carte Arduino du signal analogique fourni ADC0.
Ci-contre on a extrait du schéma structurel complet de la carte Tick Tock la partie correspondant à la thermistance.



À l'aide d'une pointe de touche et d'un voltmètre :

- Relever la tension V_{ADC0} à température ambiante : V_{ADC0} = ……..
- En chauffant légèrement la thermistance avec le décapeur thermique (attention à ne pas trop chauffer), vérifiez que la tension V_{ADC1} varie.
- Relever la nouvelle valeur : ……
  - La tension augmente-t-elle ou diminue-t-elle ? .....
  - En déduire si la thermistance est à coefficient de température positif (CTP) ou négatif (CTN). Justifier.

……
……
……

C) ÉLABORATION DE PROGRAMMES

C.1) Programme « test3A »

1) Proposer un programme pour afficher toutes les secondes, sur le moniteur intégré à l'EDI Arduino, les valeurs successives du nombre donné par le convertisseur analogique numérique du microcontrôleur grâce à la fonction **`analogRead()`**.

Visa du prof ☐

C2) Programme « test3B »

Il s'agit d'afficher sur le moniteur intégré à l'EDI Arduino directement la température en °C toutes les secondes.

Exemple :
```
temperature = 25°C
temperature = 26°C
temperature = 26°C
```

La relation suivante permet de déterminer la valeur de la résistance de la thermistance connaissant la valeur numérique N renvoyée par **`analogRead()`**:

```
resistance = (float)(1024-N)*10000/N;
```

`resistance` doit être déclarée en tant que float.
N est la valeur renvoyée par **`analogRead()`**.
10000 correspond à la valeur de la résistance en série avec la thermistance.

Comme on connaît la loi de variation de la résistance de la thermistance en fonction de la température, on peut calculer la température connaissant la valeur de la résistance.

Pour la thermistance utilisée, on utilise la relation suivante :

`temperature` doit être déclaré comme un float.

```
temperature = 1 / (log(resistance/10000)/3975 + 1/298.15) - 273.15;
```

👁 Valider le bon fonctionnement de votre programme.

Visa du prof ☐