

APP Electronique-Signal

Compte-rendu **n°1** :
Découverte programmation microcontrôleur



Membres : MARIOLLE Henri, PÉROU Alexandre, TARDIEU Maxime, TAYO FOYO Cabrel, TERISSE Arthur, THAYAPARAN Sangar

I. Introduction

Durant cette première mission, nous sommes amenés à prendre en main le microcontrôleur, prendre connaissance du cahier des charges, et acquérir les compétences basiques nécessaires à la réalisation des tâches. Pour ce faire, nous utiliserons le logiciel **ENERGIA** afin de programmer le microcontrôleur en langage C.

II. Questions et manipulations

1. Connecter une LED tricolore aux ports du microcontrôleur et démontrer l'affichage des couleurs suivantes

```
// definitions of the ports
#define V_LED 28
#define B_LED 27
#define R_LED 29

// initialisation
void setup() {
    // initialize the digital pin as an output.
    pinMode(V_LED, OUTPUT);
    pinMode(R_LED, OUTPUT);
    pinMode(B_LED, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {

    digitalWrite(B_LED, HIGH);
    digitalWrite(V_LED, LOW);
    digitalWrite(R_LED, LOW);
    delay(1000);

    digitalWrite(B_LED, LOW);
    digitalWrite(V_LED, LOW);
    digitalWrite(R_LED, HIGH);
    delay(1000);

    digitalWrite(B_LED, LOW);
    digitalWrite(V_LED, HIGH);
    digitalWrite(R_LED, LOW);
    delay(1000);

    digitalWrite(B_LED, LOW);
    digitalWrite(V_LED, HIGH);
    digitalWrite(R_LED, HIGH);
    delay(1000);
}
```

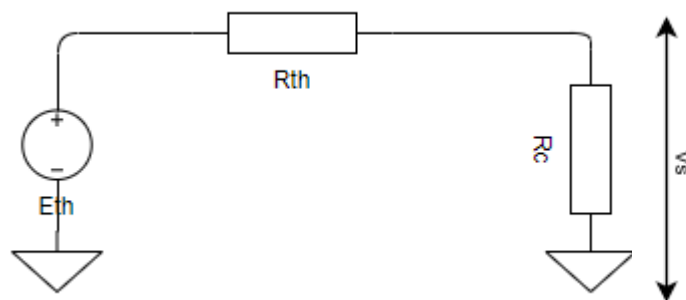
Code qui affiche les couleurs BLEU - ROUGE - VERT - JAUNE successivement.

- a. Quel courant passe dans les LEDs lorsqu'elles sont allumées ?
Comment maîtriser ce courant ?

Le courant passant dans les LED est un courant continu et polarisé. Il est possible de le maîtriser à l'aide d'une résistance branché en série.

- b. Quel est le courant maximum que peut fournir le port de sortie digital lorsqu'il est au niveau haut ?

La tension lorsque le courant est au niveau haut est comprise entre 2V et 3,3V. A l'aide d'un montage simple (voir ci-dessous), on mesure une tension de 2,1 V au borne de notre résistance de 220Ω.



Montage permettant de mesurer la tension en simplifiant le microcontrôleur à l'aide de l'équivalence Thevenin.

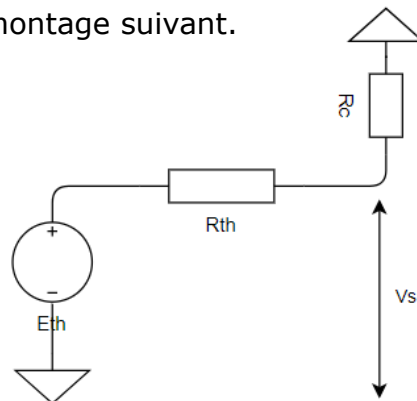
La formule du pont diviseur de tension nous permet de déduire le calcul suivant :

$$V_s = \frac{R_c}{R_{th} + R_c} * E_{th} \Leftrightarrow R_{th} = R_c * \left(\frac{E_{th}}{V_s} - 1 \right)$$

On obtient alors $R_{th} = 126\Omega$. Pour obtenir le courant maximal, il faut prendre le cas où V_s est minimal car $I_{max} = \frac{E_{th} - V_s}{R_{th}}$, d'après la loi des Mailles. Avec $E_{th} = 3,3V$ et $V_s = 2V$. On obtient un **courant maximal de 10,3 mA** dans le port de sortie digital au **niveau haut**.

- c. Quel est le courant maximum que peut fournir le port de sortie digital lorsqu'il est au niveau bas ?

Lorsque le port de sortie est au niveau bas, la tension est comprise entre 0 et 0,4V. Pour mesurer le courant maximum du port de sortie du microcontrôleur, on mesure la tension dans le montage suivant.



On obtient $V_{Rc} = 2,29V$, et $V_{Rc} = R * I \Leftrightarrow I = \frac{V_{Rc}}{Rc} = 10,2mA$. Ensuite on détermine $V_s = R_{th} * I \Leftrightarrow R_{th} = \frac{3,3 - 2,29}{0,0102} = 99\Omega$. Pour trouver le maximum, on prend $V_{sl} = 0,4V = R_{th} * I_{max} \Leftrightarrow I_{max} = \frac{V_{sl}}{R_{th}} = \frac{0,4}{99} = 4mA$.

2. Connexion du potentiomètre à une entrée analogique du microcontrôleur

```

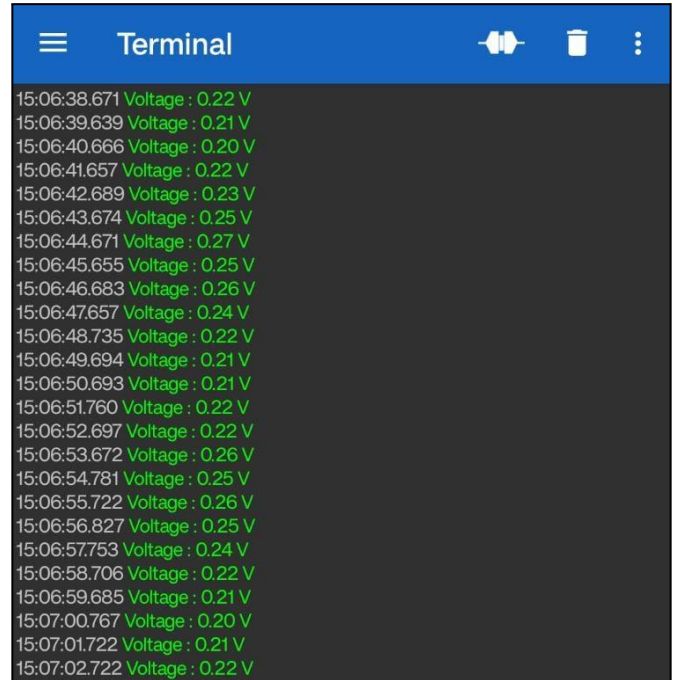
*/
int sensorPin = A9;    // select the input pin
int sensorValue = 0;  // variable to store the

void setup() {
  // declare the ledPin as an OUTPUT:
  Serial.begin(9600);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);

  Serial.print("La valeur du capteur est : ");
  Serial.print(sensorValue*3.3/4096);
  Serial.println(" V");
  delay(1000);
}

```



Terminal

```

15:06:38.671 Voltage : 0.22 V
15:06:39.639 Voltage : 0.21 V
15:06:40.666 Voltage : 0.20 V
15:06:41.657 Voltage : 0.22 V
15:06:42.689 Voltage : 0.23 V
15:06:43.674 Voltage : 0.25 V
15:06:44.671 Voltage : 0.27 V
15:06:45.655 Voltage : 0.25 V
15:06:46.683 Voltage : 0.26 V
15:06:47.657 Voltage : 0.24 V
15:06:48.735 Voltage : 0.22 V
15:06:49.694 Voltage : 0.21 V
15:06:50.693 Voltage : 0.21 V
15:06:51.760 Voltage : 0.22 V
15:06:52.697 Voltage : 0.22 V
15:06:53.672 Voltage : 0.26 V
15:06:54.781 Voltage : 0.25 V
15:06:55.722 Voltage : 0.26 V
15:06:56.827 Voltage : 0.25 V
15:06:57.753 Voltage : 0.24 V
15:06:58.706 Voltage : 0.22 V
15:06:59.685 Voltage : 0.21 V
15:07:00.767 Voltage : 0.20 V
15:07:01.722 Voltage : 0.21 V
15:07:02.722 Voltage : 0.22 V

```

Code affichant la valeur numérique et la tension équivalente sur le port analogique à l'aide de Putty.

3. Mesure de la consommation de la carte

Pour déterminer la consommation de la carte, on a mesuré le courant passant dans le microcontrôleur, et on a obtenu un **ampérage de 0,12A**. Par ailleurs, la tension émanant de l'alimentation USB est de **5V**. Après application de la loi d'Ohm, on obtient une **puissance de 0,6 Wh**.

Test de consommation de la carte CeMEQ et équivalent carbone:

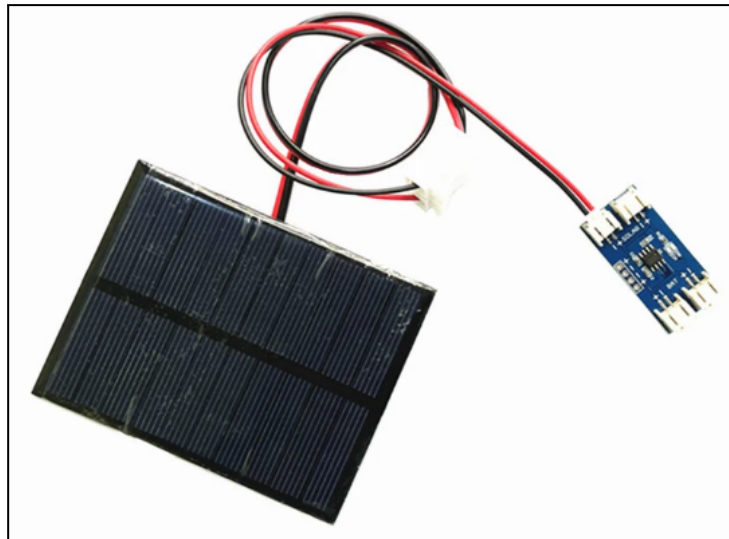
Conditions:

- Batterie de 99,6Wh
- Carte allumée en continu
- Utilisation d'énergie nucléaire uniquement: 12gCO₂/kWh

Résultats:

- Temps d'utilisation sur batterie = **166h**
- **1** utilisation complète de la batterie émet **7,2 mg de CO₂**

Concernant le panneau solaire, nous avons opté pour le modèle suivant capable d'alimenter notre carte (voir photo). Ses caractéristiques sont les suivantes: tension de 5.5V, courant de 200mA, et une puissance maximum d'émission de 1W (supérieur au 0,6W nécessaire au fonctionnement de notre carte).



4. Explication des paramètres du microcontrôleur

a. Les différentes mémoires

- **256ko de mémoire flash monocycle.** Ce type de mémoire est non volatile, elle conserve les données même lorsque l'alimentation est coupée. La mémoire flash est composée de cellules de mémoire qui sont organisées en blocs. Chaque cellule stocke une petite quantité de données sous forme de charge électrique. La mémoire flash est préférée pour sa capacité de stockage élevée, sa vitesse de lecture et d'écriture rapide, et sa faible consommation d'énergie.

- **32ko de mémoire SRAM à cycle unique.** Celle-ci est un type de mémoire vive utilisé pour stocker des données de manière temporaire. Contrairement à la DRAM qui nécessite des rafraîchissements périodiques pour conserver les données, la mémoire SRAM les conserve tant qu'elle est alimentée en électricité. Par ailleurs, elle a un temps d'accès plus court mais consomme plus d'énergie que la mémoire DRAM.

- **Mémoire ROM interne** chargée avec le logiciel d'exploitation TivaWare. Cette mémoire permet de stocker des données de manière permanente.

Lors de la fabrication, les données sont stockées dans la mémoire ROM et ne peuvent être modifiées, d'où l'appellation Read-Only Memory. Plusieurs types de mémoire ROM existent comme l'EEPROM (voir page suivante).

- **2ko de mémoire EEPROM.** Les données stockées dans une EEPROM peuvent être modifiées par l'utilisateur. Elle est généralement utilisée pour stocker des données importantes qui nécessitent une certaine flexibilité comme celles de la configuration du système.

b. Vitesse de l'horloge

La vitesse d'horloge mesure le nombre de cycles que votre CPU exécute par seconde. Il est mesuré en Hz (hertz). D'après la documentation, celle-ci est de 80MHz.

c. L'intérêt d'une unité de calcul flottant

Le microcontrôleur possède cette fonctionnalité. Celle-ci permet d'augmenter la précision et la rapidité lors des opérations basiques, et limite donc les erreurs d'arrondis. Le calcul des nombres flottants utilise une logique très différente des opérations décimales, ce qui rend les processeurs inefficaces. Une unité de calcul des nombres flottants permet alors d'accélérer l'exécution des programmes plus complexes.

d. Les types de périphériques embarqués et leurs usages

Sur le microcontrôleur fourni, on retrouve une multitude de périphériques. Voici une liste exhaustive :

1. Entrée alimentation électrique
2. Entrée alimentation sur batterie pour une utilisation sans contrainte.
3. Module bluetooth relié au port série 1 de la carte TIVA
4. 1 prise jack OUT permettant de connecter un casque.
5. 1 prise jack IN reliée au port analogique IN de la carte TIVA permettant de connecter un microphone.
6. Un convertisseur Digital/Analogique qui permet de transformer un signal numérique en un signal continu
7. 2 entrées permettant l'ajout d'une commande de servomoteurs.
8. 3 connecteurs pour pouvoir supporter des cartes complémentaires.

5. Connexion au CeMEQ à l'aide du Bluetooth

```
int sensorPin = A9;    // select the input pin for the potentiometer
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  Serial1.begin(9600);
}

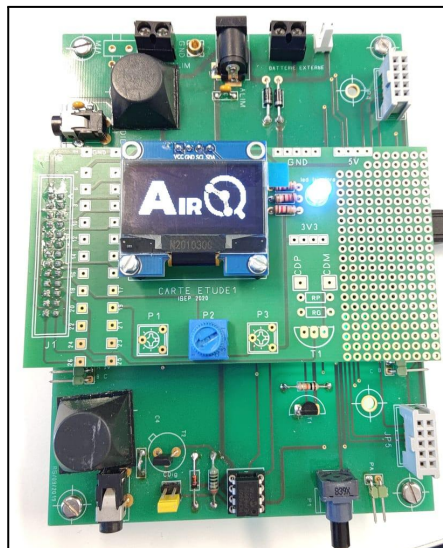
void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);

  Serial1.print("Voltage : ");
  Serial1.print(sensorValue*3.3/4096);
  Serial1.println(" V");
  delay(1000);
}
```

Code permettant d'afficher sur téléphone la tension sur le port analogique à l'aide du Bluetooth et la fonction serial 1.

6. Câblage de l'afficheur OLED et affichage de message

A partir du dossier fourni par l'ISEP, nommé test screen nous n'avons eu qu'à convertir notre logo au format de l'écran (128 x 64 px) et utiliser l'utilitaire image2cpp disponible à l'adresse suivante: <https://javl.github.io/image2cpp/>. Il nous a fallu ensuite remplacer le logo de l'isep se trouvant dans le dossier par notre logo stocké au format hexadécimal, et téléverser le programme.



Carte avec afficheur OLED affichant le logo AirQ