

# Circuit Playground Express

## Systèmes embarqués



### Objectif :

*Circuit Playground Express  
Micro Python*

Découvrir quelques aspects de la mise en œuvre de systèmes embarqués autour d'une carte peu onéreuse aux multiples possibilités la Playground Express d'Adafruit associée au langage micro python.

La carte est pilotée grâce à un microcontrôleur 32 bits ATMEL ATSAMD21, doté d'un cœur ARM® Cortex®-M0+ processor, opérant à une fréquence de 48 MHz.

### **SMART ARM-Based Microcontroller**

*(Atmel est une branche de Microchip depuis son rachat en 2012 mais a conservé ses designs de circuits compacts et performants.)*

### Quelques indications :



Une notion à retenir.

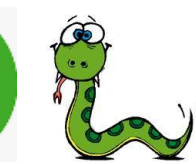
Attention requise danger pour le matériel.



## **Sommaire**

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>TP_1 : Entrées et sorties Digital IN et OUT.....</b> | <b>2</b> |
| 1.1      | Objectif :.....   | 2        |
| 1.2      | Principe du fonctionnement.....                         | 2        |
| 1.3      | Étude de l'entrée.....                                  | 3        |
| 1.4      | Configuration logicielle de l'entrée. ....              | 3        |
| 1.5      | Le schéma de la carte.....                              | 4        |
| <b>2</b> | <b>Les premiers scripts .....</b>                       | <b>4</b> |
| 2.1      | Essai des switch A et B .....                           | 4        |
| 2.2      | Combinaison des switches.....                           | 4        |
| <b>3</b> | <b>Et pour la led.....</b>                              | <b>5</b> |
| 3.1      | Le schéma.....  | 5        |
| 3.2      | Comment allumer la led ?.....                           | 5        |
| 3.3      | Détermination de valeurs de résistances.....            | 5        |





# 1 TP\_1 : Entrées et sorties Digital IN et OUT

## 1.1 Objectif :

Réaliser une première acquisition d'une entrée et répercuter sa valeur pour commander l'allumage d'une led.

Découvrir comment le processeur Atmel utilise, avec son programme  $\mu$ Python, les signaux électriques présents sur ses broches d'entrées, en mode logique, puis actionne ses broches de sorties en conséquence.

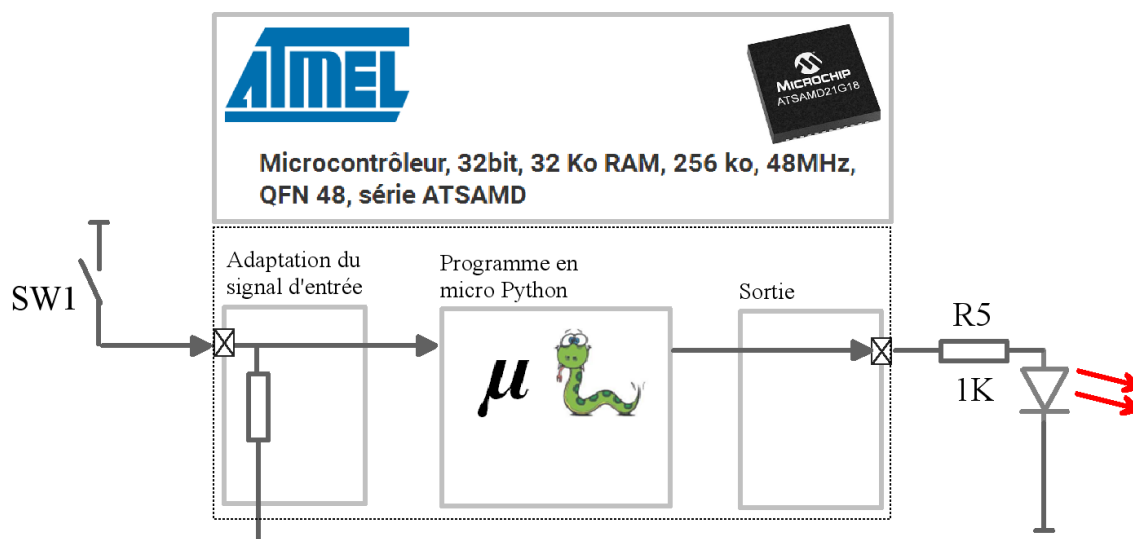
Découvrir les liens entre le schéma électronique de la carte et le dimensionnement de quelques composants électroniques associés, ici résistance et led.

Circuit Playground Express  
Micro Python



## 1.2 Principe du fonctionnement

Le synoptique ci-dessous présente le fonctionnement général



Q1. Décrire succinctement le fonctionnement attendu.

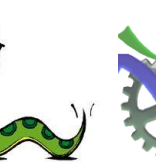
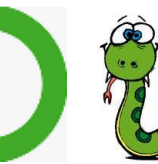
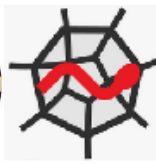
Q2. Combien y a-t-il de valeurs différentes pour un signal logique ?

Q3. Comment sont notées ces valeurs ?



Dans un circuit électronique ces deux valeurs sont le plus souvent représentées par deux niveaux de tensions différents. Par exemple 0V et 5V, ou 0V et 3.3V. On parle de niveau haut et niveau bas.

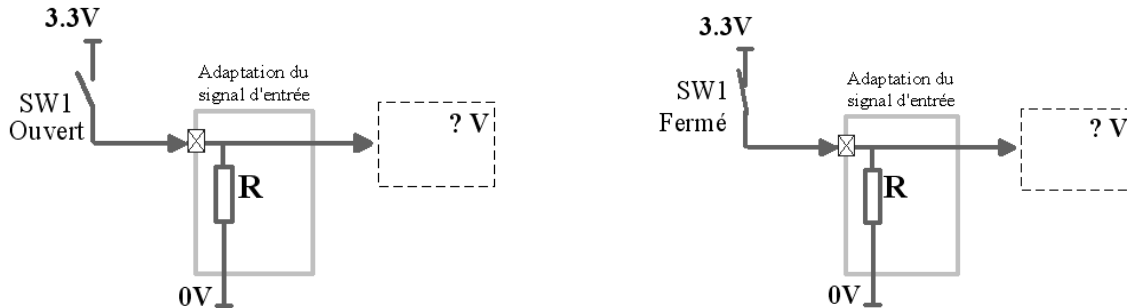




### 1.3 Étude de l'entrée

L'entrée a pour objectif de transformer les deux positions de l'interrupteur SW1 en deux niveaux de tensions haut et bas. Les deux situations sont représentées ci-dessous :

Q4. Indiquez les valeurs présentes sur la broche d'entrée dans les cases en pointillées.  
Choix possibles : 3.3V, 0V.



Q5. Indiquez les niveaux logiques obtenus dans le tableau ci-dessous :

| Position de l'interrupteur SW1 | Valeur de la tension sur la broche | Niveau logique obtenu en entrée |
|--------------------------------|------------------------------------|---------------------------------|
| Ouvert                         |                                    |                                 |
| Fermé                          |                                    |                                 |

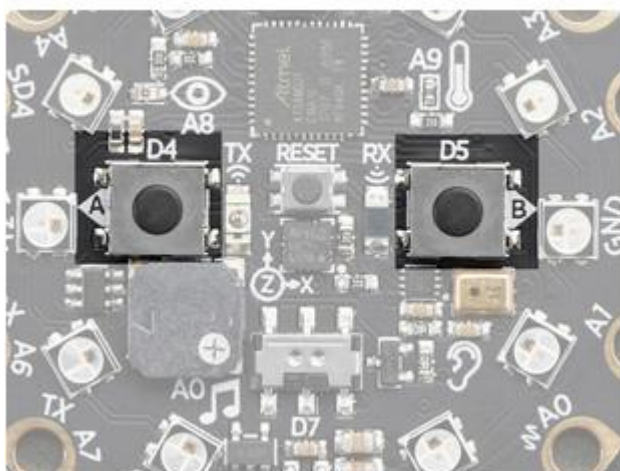
### 1.4 Configuration logicielle de l'entrée.

Nous voyons que la résistance R, obligatoire pour fixer la valeur de l'entrée quand le switch SW1 est ouvert est représentée à l'intérieur du SOC.

Voilà les deux lignes de codes :

```
# Désignation de la broche utilisée
button = digitalio.DigitalInOut(board.BUTTON_A)
```

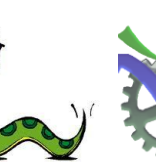
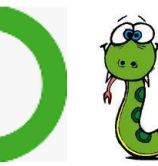
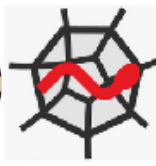
```
# Configuration de la résistance de tirage pulldown
button.switch_to_input(pull=digitalio.Pull.DOWN)
```



There are two large **A** and **B** buttons, connected to digital #4 (Left) and #5 (Right) each. These are unconnected when not pressed, and connected to 3.3V when pressed, so they read HIGH. Set the pins #4 and #5 to use an internal pull-down resistor when reading these pins so they will read LOW when not pressed.

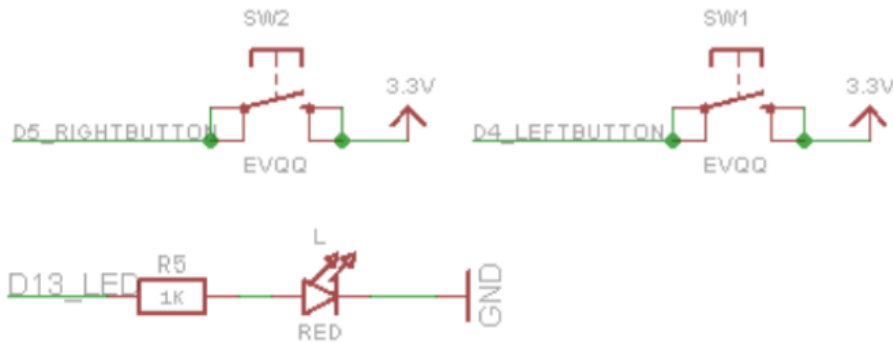
<https://learn.adafruit.com/adafruit-circuit-playground-express/guided-tour>





## 1.5 Le schéma de la carte

Le schéma des switches BUTTON\_A et BUTTON\_B et de la led D13



## 2 Les premiers scripts

### 2.1 Essai des switch A et B

Q6. Proposez les deux lignes de codes à écrire pour utiliser la BUTTON\_B

```
buttonB = digitalio.DigitalInOut(board.BUTTON_B)
buttonB.switch_to_input(pull=digitalio.Pull.DOWN)
```



Script1. Faire fonctionner le script CPX\_TP1\_Digital IN OUT.py

```
# for Adafruit Industries
#
# SPDX-License-Identifier: MIT
# Circuit Playground digitalio example

import time
import board
import digitalio

led = digitalio.DigitalInOut(board.D13)
led.switch_to_output()

button = digitalio.DigitalInOut(board.BUTTON_A)
button.switch_to_input(pull=digitalio.Pull.DOWN)

while True:
    if button.value: # button is pushed
        led.value = True
    else:
        led.value = False
    time.sleep(0.01)
```



Script2. Adapter le script précédent pour prendre en compte le switch B.

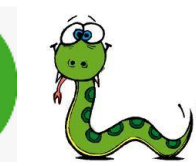


### 2.2 Combinaison des switches



Script3. Adapter le script précédent pour allumer la led quand les deux switches A et B sont appuyés.



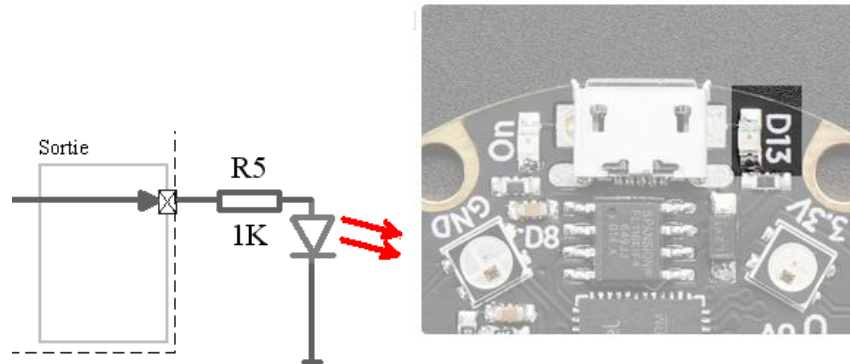


### 3 Et pour la led

#### 3.1 Le schéma

Le microcontrôleur présente deux valeurs de tension sur une broche de sortie logique à savoir :

- Un niveau haut représenté par une tension de 3.3V
- Un niveau bas représenté par une tension de 0V

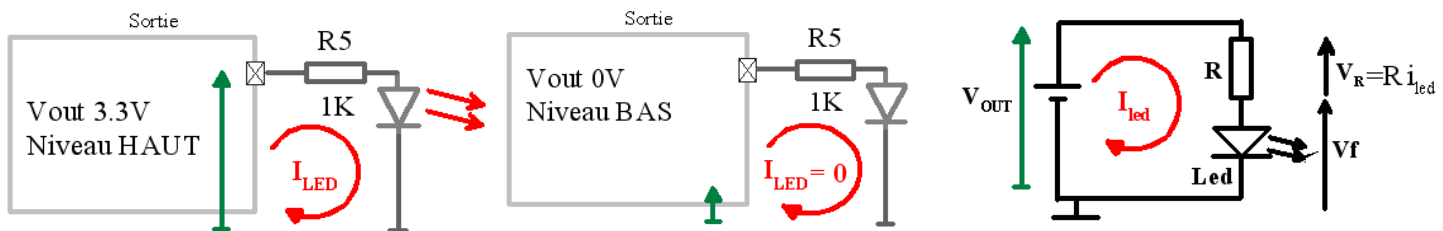


#### Red #13 LED

To the right of the USB connector. This LED does double duty. Its connected with a series resistor to the digital #13 GPIO pin. It pulses nicely when the CPX is in bootloader mode, and its also handy for when you want an indicator LED. Many first projects blink this LED to prove that programming worked.

#### 3.2 Comment allumer la led ?

Deux valeurs de tensions de sortie donc deux schémas à étudier :



C'est la valeur de la résistance qui fixera le courant traversant la led quand  $V_{out} = 3.3V$  voilà les relations :

$$R = \frac{V_{out} - V_f}{i_{led}} \quad i_{led} = \frac{V_{out} - V_f}{R}$$

Q7. Déterminer le courant  $i_{led}$  traversant la led sur la carte CPX (Avec  $V_f = 1.8V$ )

#### 3.3 Détermination de valeurs de résistances

Q8. Avec une tension de commande de 3.3V déterminer les valeurs de résistances pour faire fonctionner la led dans les conditions ci-dessous :

| LEDS  | ROUGE | VERTE | ORANGE |          |
|-------|-------|-------|--------|----------|
| VF    | 1,7   | 2,1   | 2,1    | V        |
| I Led | 10    | 10    | 10     | mA       |
| R     |       |       |        | $\Omega$ |

