

# TP4 - TinkerCAD

Théodore BONDON, Gabin DIETSCH, Nicolas GROUSSEAU, Rémi VAN BOXEM

March 8, 2021

## Contents

<b>1</b>	<b>Arduino : le composant</b>	<b>1</b>
1.1	Avec une breadboard . . . . .	1
1.1.1	Résistance et LEDs . . . . .	1
1.1.2	Tableau . . . . .	2
<b>2</b>	<b>Arduino : le code</b>	<b>2</b>
2.1	Allumer une LED avec sortie logique et analogique . . . . .	2
2.1.1	Valeur de la résistance . . . . .	2
2.1.2	Pins analogique . . . . .	2
2.1.3	Valeurs analogiques . . . . .	2
2.1.4	Code de la fonction decrease . . . . .	2
2.2	Ajouter un bouton en entrée . . . . .	3
2.2.1	Code de déclaration . . . . .	3
2.2.2	Pins permettant une interruption . . . . .	3
2.2.3	Modification de la fonction decrease . . . . .	3

## 1 Arduino : le composant

### 1.1 Avec une breadboard

#### 1.1.1 Résistance et LEDs

Valeur	$20p\Omega$	$200m\Omega$	$20\Omega$	$200\Omega$	$20K\Omega$	$2M\Omega$	$2G\Omega$
Allumée ?	Fortement	Fortement	Fortement	Fortement	Faiblement	Faiblement	Très faible
Danger ?	Oui	Oui	Oui	Non	Non	Non	Non

### 1.1.2 Tableau

Numéro	Utile	Pas d'effet
1		x
2	x	
3		x
4	x	
5	x	
6		x
7	x	
8	x	
9	x	
10		x
11		x

## 2 Arduino : le code

### 2.1 Allumer une LED avec sortie logique et analogique

#### 2.1.1 Valeur de la résistance

La résistance qui permet de faire briller la LED le plus vivement possible sans l'endommager parmi les résistances proposées précédemment est la résistance de  $200\Omega$ .

#### 2.1.2 Pins analogique

Les pins qui permettent à la LED de s'allumer sont les pins **3, 5, 6, 9, 10** et **11**.

#### 2.1.3 Valeurs analogiques

Valeur	État
512	Éteint
511	Allumé (fort)
257	Allumé (faible)
256	Éteint
100	Allumé (fort)
1	Allume (faible)

#### 2.1.4 Code de la fonction decrease

```
void decrease(int pin, int timems, int stepNumber) {
```

```

    int volt_step = 512 / stepNumber;
    int ETA = 511;
    for (int i = 0; i <= stepNumber; i++) {
        analogWrite(pin, ETA);
        ETA = ETA - volt_step;
        delay(timems);
    }
}

```

## 2.2 Ajouter un bouton en entrée

### 2.2.1 Code de déclaration

```

void setup()
{
    pinMode(2, INPUT);
    pinMode(11, OUTPUT);
}
void loop()
{
    if (digitalRead(2) == HIGH) {
        decrease(11, 1000, 10);
    }
}

```

### 2.2.2 Pins permettant une interruption

Les pins permettant une interruption sont les Pins **1** et **2**.

### 2.2.3 Modification de la fonction decrease

Reprenez votre fonction decrease mais retirez l'argument timems pour en faire une variable globale. Utilisez cette fonction afin de rallumer puis éteindre progressivement la led de manière répétée.

Rattachez le bouton à une interruption qui permet de changer cette variable timems pour passer d'une valeur à une autre en cycle entre trois modes lent(1sec) modéré(500ms) et rapide(100ms).

```

int lent = 1000;
int moderate = 500;
int rapide = 100;
int timems = lent;

```

```

void decrease(int pin, int stepNumber) {
    int volt_step = 512 / stepNumber;
    int ETA = 511;
    for (int i = 0; i <= stepNumber; i++) {
        analogWrite(pin, ETA);
        ETA = ETA - volt_step;
        delay(timems);
    }
}

void setup()
{
    //pinMode(2, INPUT);
    pinMode(11, OUTPUT);
    attachInterrupt(digitalPinToInterrupt(2), change_value, FALLING);
}

void change_value(){
    if (timems == lent){
        timems = moderate;
        return;
    }
    else if (timems == moderate){
        timems = rapide;
        return;
    }
    else if (timems == rapide){
        timems = lent;
        return;
    }
}

void loop()
{
    decrease(11, 10);
}

```