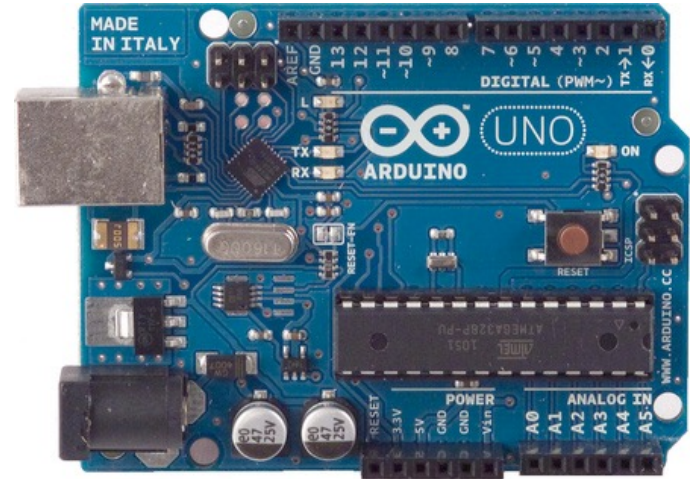


Arduino



Les bases

Pierre Rossel
2021

Setup et loop

- Un programme arduino est composé au moins de 2 fonctions
 - Setup
 - Exécutée une seule fois au début du programme
 - Loop
 - Exécutée en boucle

Sortie

- `pinMode(13, OUTPUT)`
 - Définit que la broche 13 doit être utilisée comme sortie
- `digitalWrite(13, HIGH)`
 - Envoie 5V sur la broche 13
- `digitalWrite(13, LOW)`
 - Envoie 0V sur la broche 13

Exemple: allumer la LED

```
// Allume la LED sur la broche 13
void setup() {
    pinMode(13, OUTPUT);
    digitalWrite(13, HIGH);
}

void loop() {
}
```

Loop

- Fonction loop() est appelée en boucle
- Exemple

```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

Constante

- Permet de donner un nom et une signification à une valeur
- La valeur ne peut pas changer pendant l'exécution
- Permet de d'isoler les valeurs numériques (la configuration) à un seul endroit

```
#define LED 13
```

```
void setup() {  
    pinMode(LED, OUTPUT);  
}
```

Variable

- Permet d'attribuer un nom à une valeur
- La valeur peut changer pendant l'exécution du programme
- La valeur est utilisée à la place du nom
- Exemple

```
#define LED 13
boolean val = HIGH;
void setup() {
    pinMode(LED, OUTPUT);
    digitalWrite(LED, val);
}
```

Entrée

- Lire une valeur numérique (pas digitale)

```
#define BUTTON 8  
pinMode(BUTTON, INPUT);  
boolean val = digitalRead(BUTTON);
```

- Etats possibles

- ❑ LOW (0V)
- ❑ HIGH (5V)

Entrée

■ Exemple

```
#define LED 13
#define BUTTON 8

void setup() {
    pinMode(LED, OUTPUT);
    pinMode(BUTTON, INPUT);
}

void loop() {
    boolean val = digitalRead(BUTTON);
    digitalWrite(LED, val);
}
```

Entrée

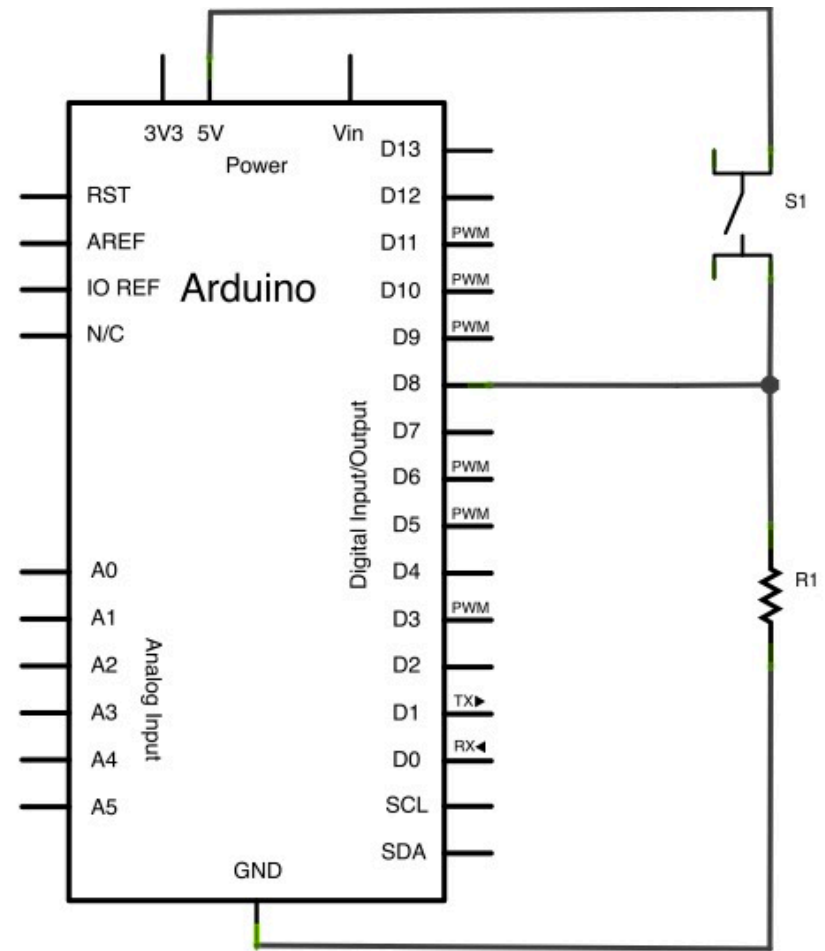
- Connecter un fil entre la broche 8 et 5V
 - la LED s'allume
- Déconnecter le fil de 5V et le mettre sur GND
 - La LED s'éteint
- Si le fil n'est pas branché, l'état de la LED n'est pas défini et peut même changer tout seul

Pull-down

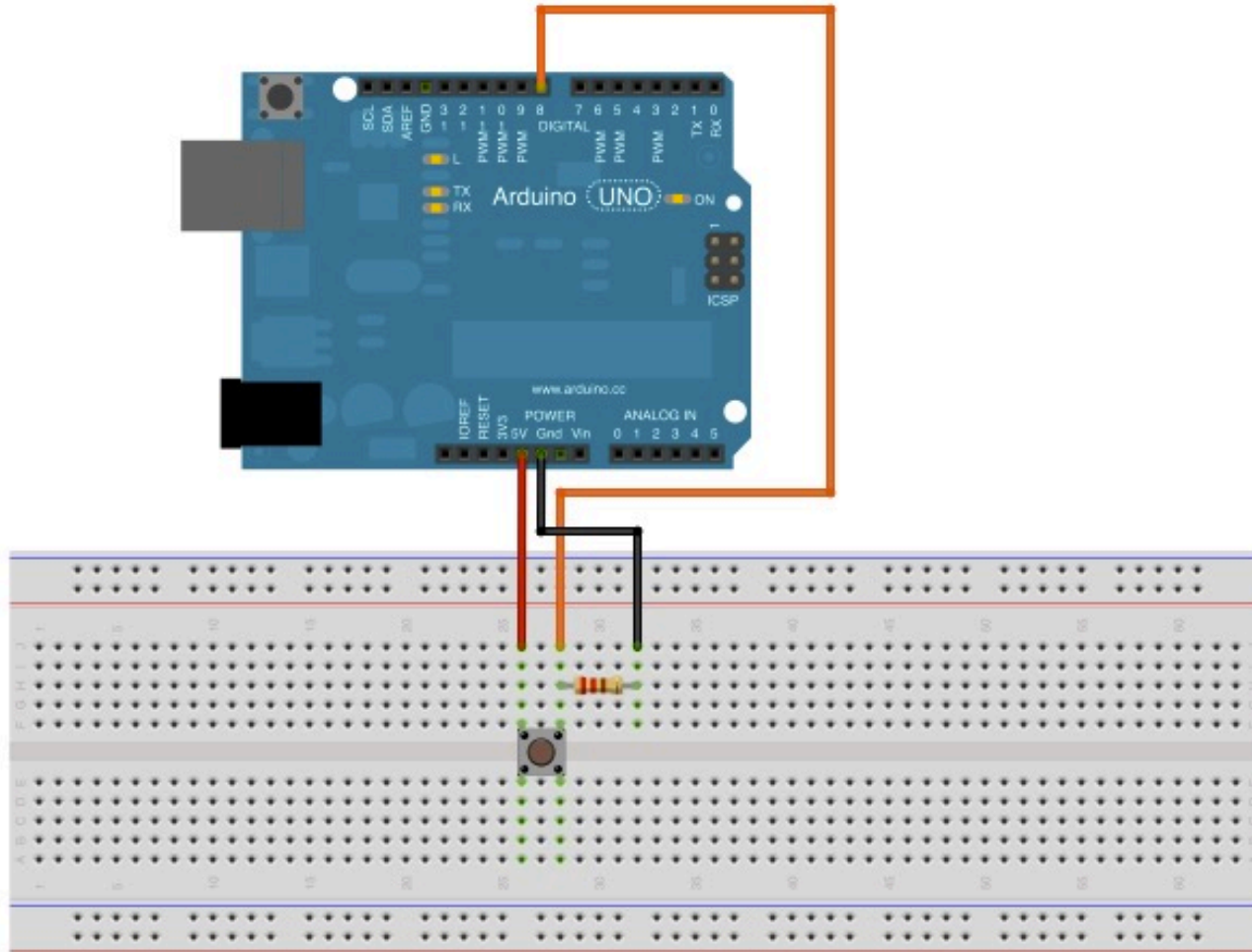
- Une résistance de 1MOhm entre l'entrée 8 et GND permet de tirer l'entrée à 0V (pull-down)
- Si on veut faire passer l'entrée 8 à HIGH, on la connecte à 5V avec le fil
 - Un petit courant passe par la résistance, mais l'entrée est bien à 5V
- On peut remplacer le fil par un bouton
 - En appuyant, l'entrée est HIGH, sinon elle est LOW

Pull-down

- R1 tire D8 vers le bas (0V) quand S1 est ouvert
- Quand S1 est fermé, D8 est directement connecté à 5V

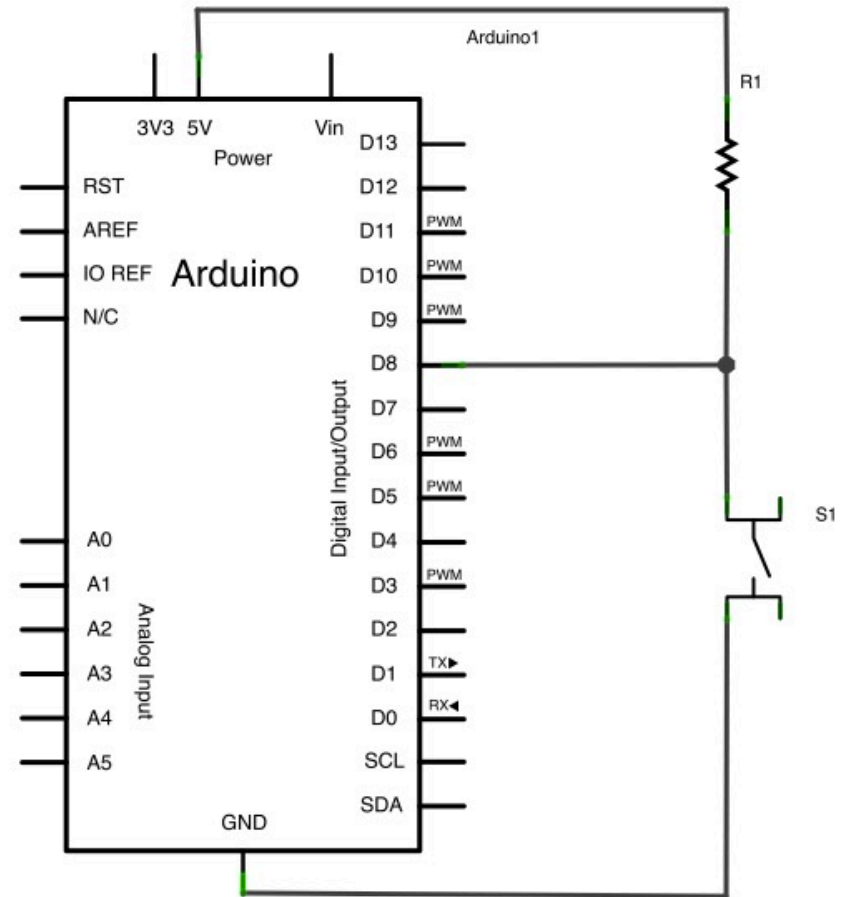


Pull-down



Pull-up

- Logique inversée
- La résistance « pull-up » tire l'entrée vers le haut (5V / HIGH) par défaut
- Un bouton permet de connecter l'entrée à 0V (LOW)



Pull-up intégrée

- Arduino dispose d'une résistance pull-up intégrée (20K Ω)
- Pour l'activer
 - `pinMode(8, INPUT_PULLUP)`
- Plus besoin d'une résistance externe
- Le bouton doit être connecté à 0V
- Attention logique inversée
 - HIGH par défaut
 - LOW quand le bouton est appuyé

Communication série

- Arduino peut communiquer avec l'ordinateur par une liaison série (via le port USB)

- Dans `setup()`

```
// initialize serial communication at 9600 bits per second:
```

```
Serial.begin(9600);
```

- Dans `loop()`

```
boolean val = digitalRead(button);
```

```
Serial.println(val);
```


Inversion

- L'opérateur « ! » permet d'inverser la valeur d'une variable

- Exemple 1

```
digitalWrite(led, !val);
```

- Exemple 2

```
void setup() {  
    Serial.begin(9600);  
    boolean val = HIGH;  
    Serial.println(val);  
    val = !val;  
    Serial.println(val);  
}
```

Dimmer manuel

```
#define LED 13
int lum = 15; // luminosité entre 0 et 20

void setup() {
    pinMode(LED, OUTPUT);
}

void loop() {
    digitalWrite(LED, HIGH);
    delay(lum);
    digitalWrite(LED, LOW);
    delay(20 - lum);
}
```

Dimmer avec bouton

```
// Lire l'état du bouton
boolean val = digitalRead(BUTTON);

// Modifie l'intensité
if (val) {
    lum = lum + 0.1;
    if (lum > 20) {
        lum = 0.0;
    }
}
```

SOS

■ Envoyer le signal SOS en morse

```
// Ti  
digitalWrite(LED, HIGH);  
delay(200);  
digitalWrite(LED, LOW);  
delay(200);
```

```
// Fin de lettre  
delay(400);
```

```
// Ta  
digitalWrite(LED, HIGH);  
delay(600);  
digitalWrite(LED, LOW);  
delay(200);
```

Fonctions

- Permettent de donner un nom à un bloc de code
- Evite la duplication de code identique

```
ti(); ti(); ti();
```

```
void ti(){  
    digitalWrite(ledPin, HIGH);  
    delay(iTiLength);  
    digitalWrite(ledPin, LOW);  
    delay(iTiLength);  
}
```