ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Gestion du temps : gérer les entrées

Comprendre les Microcontrôleurs

Jean-Daniel NICOUD et Pierre-Yves ROCHAT

Gestion du temps : gérer les entrées

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

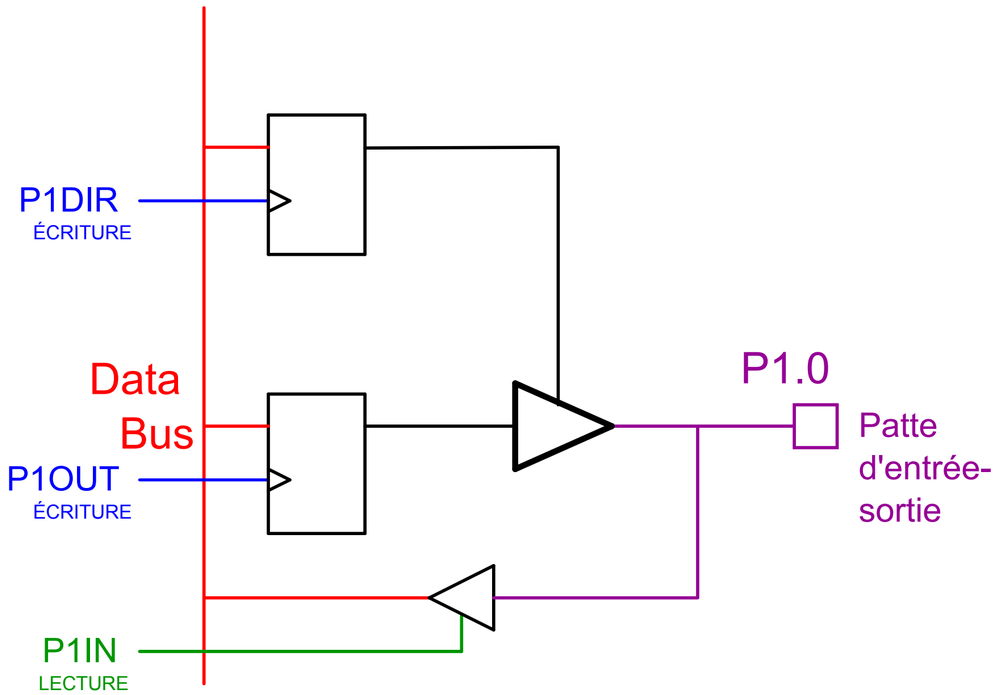
- Incertitude des entrées
- Scrutation (*Polling*)
- Détection d'un flanc
- Rebonds de contact
- Comptage d'événements
- Mesure du temps

Incertitude des entrées

- Reprenons le schéma correspondant à une patte de microcontrôleur :

- La bascule offre une certitude sur la valeur de la sortie

- Le passeur d'entrée ne donne la valeur de l'entrée qu'à l'instant de la lecture



La scrutation et ses contraintes

- La scrutation (***polling*** en anglais) : « Examen répété de l'état d'un ou plusieurs éléments d'un système pour y détecter un changement éventuel »
- Les contraintes de temps varient d'un problème à l'autre :

La scrutation et ses contraintes

Quizz :

Quel est le temps minimum entre deux scrutations pour prendre connaissance de manière fiable de la rotation de ces deux dispositifs ?

- Un tourniquet de métro
  1 impulsion lors du passage d'une personne

  - 1 seconde
  - 1 ms
  - 1 µs

- Un moteur de moto
  12 impulsions par tour, maximum 8'000 tours/minute

  - 1 seconde
  - 1 ms
  - 1 µs

La scrutation et ses contraintes

- Le piéton va prendre environ une seconde pour faire basculer le tourniquet.
  Une scrutation chaque 10ms serait suffisante.

- Le moteur tourne au maximum à 8'000 tours par minute,
  soit 133 tours par seconde. La période minimale est donc de 7.5 ms.

  Avec 12 impulsions, le temps minimum entre deux changements est de
  7'500 µs / 24 = 312 µs
  Une scrutation toutes les ms est donc insuffisante.

Lecture d'une entrée

- Comment écrire un programme qui :

  « allume la LED rouge lorsqu'on presse sur le bouton-poussoir ».

- Une erreur de débutant :

```
while (1) { // boucle principale
    if (PoussoirOn) {
        AllumeLed;
    }
}
```

Lecture d'une entrée

- Le programme doit : « allume la LED rouge lorsqu'on presse sur le bouton-poussoir et l'éteindre le reste du temps ».

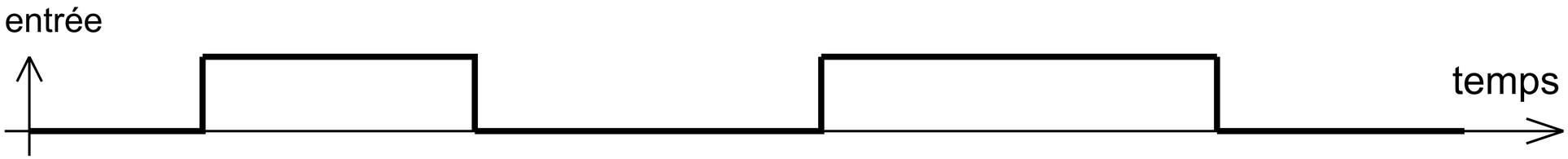
- Le programme :

```
while (1) { // boucle principale
    if (PoussoirOn) {
        AllumeLed;
    } else {
        EteintLed ;
    }
}
```

Une variante, bloquante :

```
while (1) {
    while (PoussoirOn) {
        AllumeLed;
    }
    EteintLed ;
}
```

Détection d'un flanc




```
while (1) { // boucle principale
    while (!EntreeOn) {
        // on attends le flanc montant
    }
    ... // action
    while (EntreeOn) {
        // on attends le flanc descendant
    }
}
```

Mémorisation de l'ancienne valeur

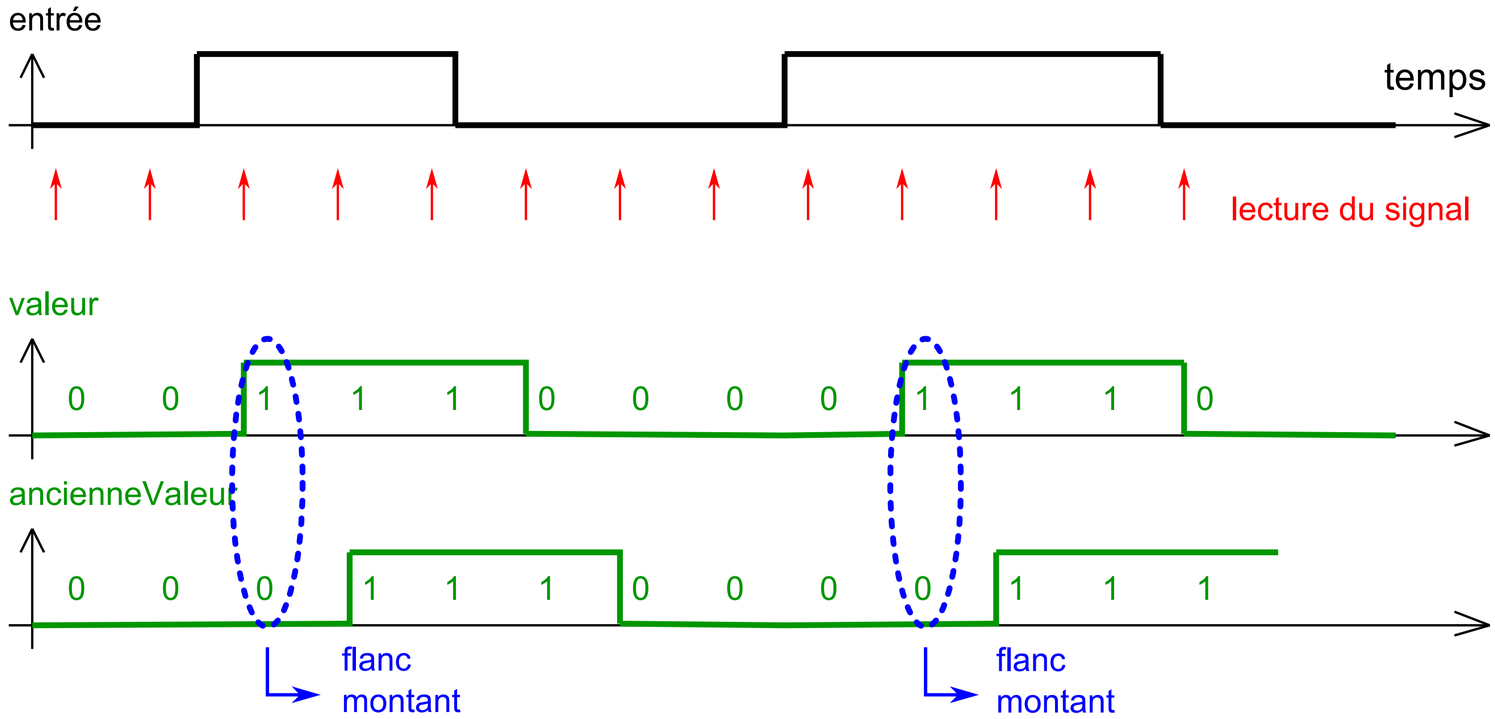
```
#define EntreeOn (digitalRead(entree))

int ancienneValeur=EntreeOn;
int valeur;

while (1) { // boucle principale
    valeur = EntreeOn;
    if (!ancienneValeur && valeur) {
      // action à effectuer sur le flanc montant
    }
    ancienneValeur=valeur;
}
```

Mémorisation de l'ancienne valeur

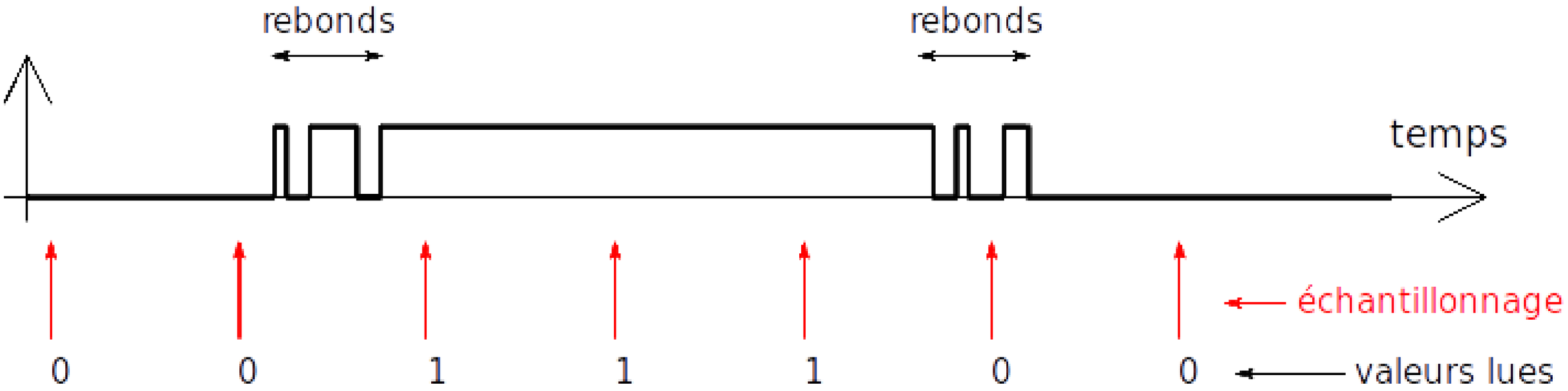
```
int ancienneValeur=EntreeOn; int valeur;
while (1) { // boucle principale
    valeur = EntreeOn;
    if (!ancienneValeur && valeur) {
      // action  sur le flanc montant
    }
    ancienneValeur
    =valeur;
}
```



Multi-tâches

```
int ancienneValeur1=Entree1On;
int valeur1;
int ancienneValeur2=Entree2On;
int valeur2;
while (1) { // boucle principale
    valeur1 = Entree1On; // Lecture les entrées
    valeur2 = Entree2On; // en début de boucle
    if (!ancienneValeur1 && valeur1) {
      // action à effectuer sur le flanc montant de l'entrée 1
    }
    if (ancienneValeur2 && !valeur2) {
      // action à effectuer sur le flanc descendant de l'entrée 2
    }
    ancienneValeur1=valeur1; // mémorisation des valeurs
    ancienneValeur2=valeur2; // en fin de boucle
}
```

Rebonds de contact



Rebonds de contact

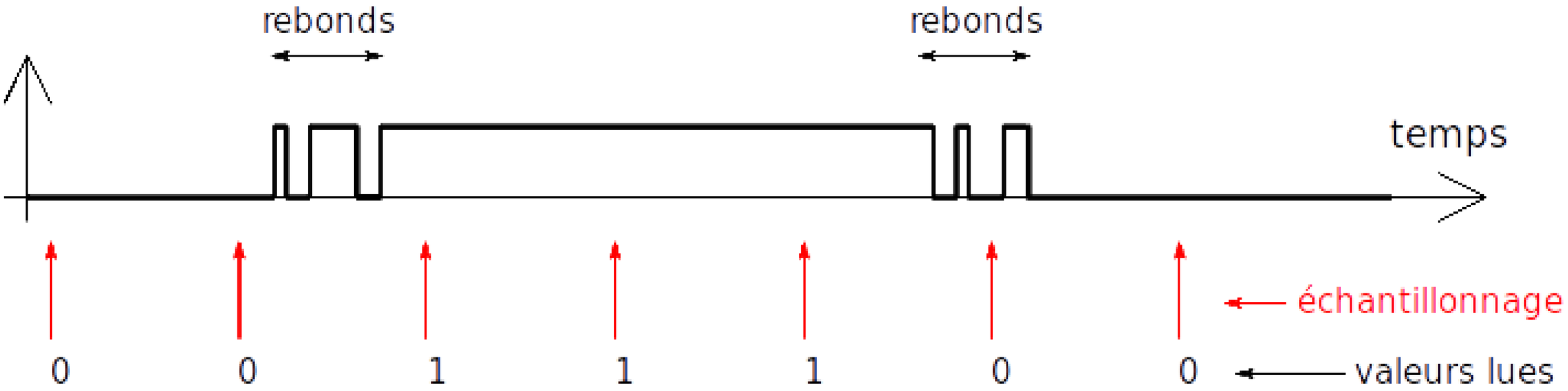
```
while (1) { // boucle principale
    valeur = digitalRead(poussoir);
    if (!ancienneValeur && valeur) {
      // action à effectuer sur le flanc montant
    }
    ancienneValeur=valeur;
    delay(20);
}
```

Rebonds de contact

```
while (1) { // boucle principale
    valeur = digitalRead(poussoir);
    if (!ancienneValeur && valeur) {
      delay(20); // masque les rebonds
      // action à effectuer sur le flanc montant
    }

    ancienneValeur=valeur;
}
```

Rebonds de contact



Rebonds de contact

```
while (1) { // boucle principale
    valeur = digitalRead(poussoir);
    if (!ancienneValeur && valeur) {
      delay(20); // masque les rebonds
      // action à effectuer sur le flanc montant
    }
     if (!ancienneValeur && valeur) {
      delay(20); // masque les rebonds
    }
    ancienneValeur=valeur;
}
```

Fréquence d'échantillonnage

- Le choix de la fréquence ne doit pas se faire au hasard

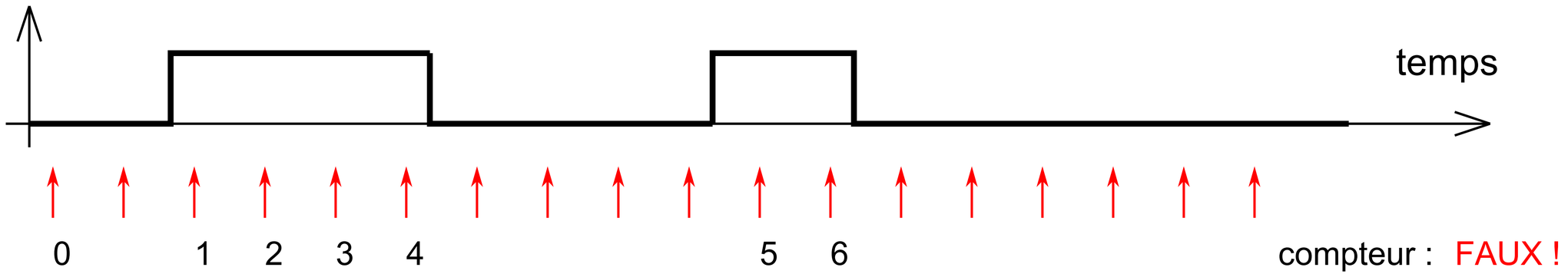
- La théorie de traitement de signal donne des indications :

$$FreqEchMin >= 2 . FreqEntréeMax$$

Comptage d'événements

- Encore un erreur de débutant :

```
while (1) { // boucle principale
    if (PoussoirOn) {
        compteur++;
    }
}
```



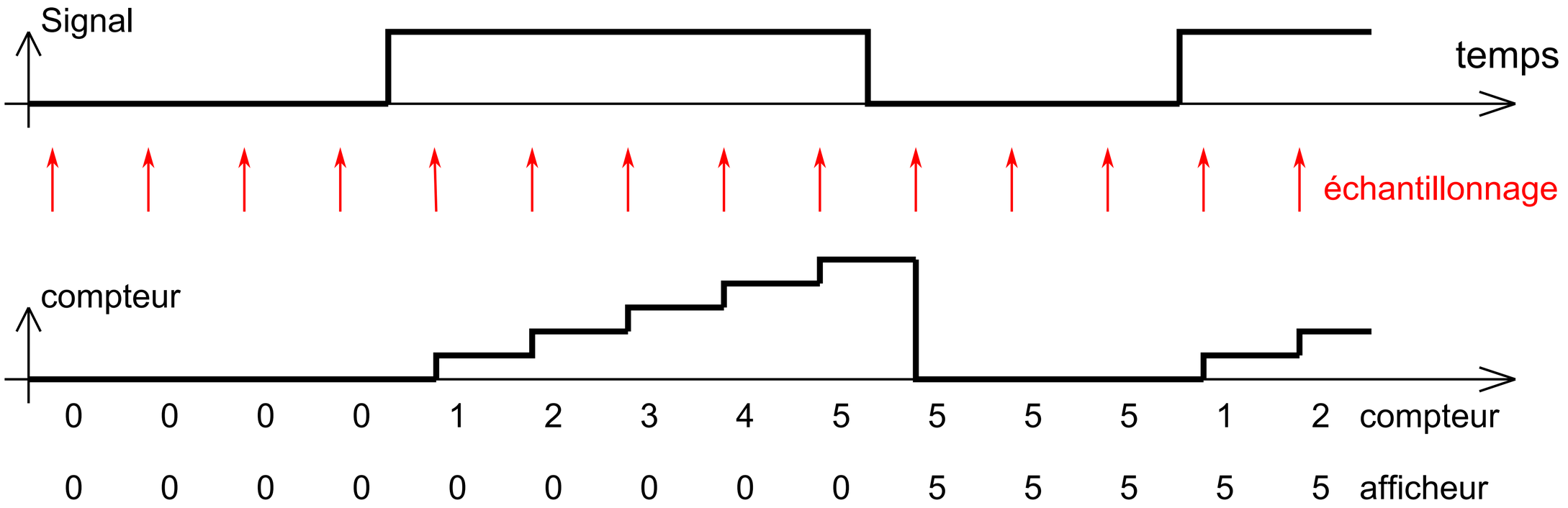
Comptage d'événements

- Appliquer la détection des flancs :

```
int ancienneValeur=EntreeOn;
int valeur;
int compteur=0;

while (1) { // boucle principale
    valeur = EntreeOn;
    if (!ancienneValeur && valeur) {
      compteur++;
    }
    ancienneValeur=valeur;
}
```

Mesure du temps



Mesure du temps

```
int ancienneValeur=EntreeOn;
int valeur;
int compteur=0;

while (1) { // boucle principale
    valeur = EntreeOn;
    if (valeur) { // entrée active ?
        compteur++; // totalise la durée de l'impulsion
    }
    if (ancienneValeur && !valeur) { // flanc descendant ?
      Affiche(compteur); // affiche la valeur
      compteur=0;
    }
    ancienneValeur=valeur;
    delay(1);
}
```

Gestion du temps : gérer les entrées

- Incertitude des entrées
- Scrutation (*Polling*)
- Détection d'un flanc
- Rebonds de contact
- Comptage d'événements
- Mesure du temps