

Architecture matérielles – Cours 4

Interruptions

IUT informatique des Cézeaux – 2014/2015

Plan

Principe général et concept

Spécificités Arduino

En pratique

Principe général des interruptions

- ▶ Évènement externe qui peut arriver à n'importe quel moment
 - ▶ Comment capturer les évènements dès qu'ils apparaissent ?
 - ▶ Dans un programme séquentiel, comment gérer les évènements externes ? (appui de boutons, détection d'un capteur, tick de timer)
- > Interruptions
- ▶ Déclenche un traitement dans le microcontrôleur
 - ▶ Traitement matériel. Très rapide

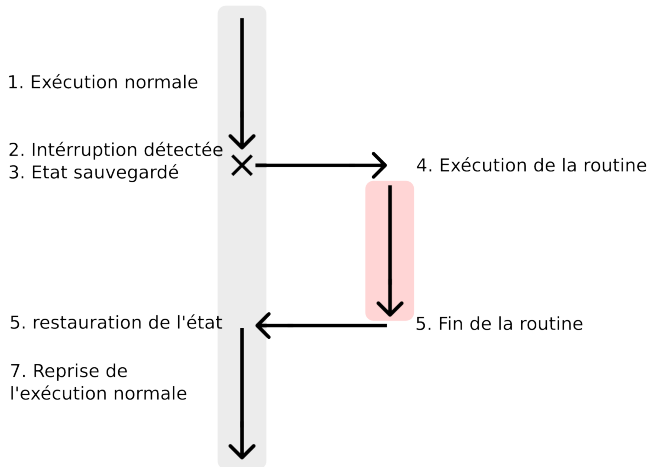
Concept général

- ▶ Le microcontrôleur exécute en permanence le même code de manière séquentielle
- ▶ Une interruption est un traitement particulier commandé par le matériel qui interrompt l'exécution habituelle du code
- ▶ Le programme s'arrête donc temporairement pour exécuter un autre programme : la routine d'interruption
- ▶ Une interruption est un système qui permet à une routine de s'exécuter suite à un évènement (matériel ou logiciel)

Concept – Séquence

- ▶ Une interruption, c'est un évènement qui déclenche les traitements suivants
 1. arrêt de l'exécution du programme courant
 2. sauvegarde de toutes les variables et de l'endroit où le programme s'est arrêté
 3. exécution de la routine
 4. restauration des variables et de l'endroit où le programme s'est arrêté
 5. reprise de l'exécution comme si rien ne s'était passé

Concept – Séquence



Priorités des interruptions

- ▶ Est-ce qu'une interruption B peut interrompre une interruption A en cours d'exécution ?
- ▶ Tous les microcontrôleurs possèdent une gestion des priorités dans les interruptions
 - ▶ C'est généralement prédéfini et non configurable. INT1 plus prioritaire que INT2...
- ? Par exemple : une arduino gère le comptage des personnes passant une porte et la détection incendie. Qui doit être le plus prioritaire ?

Plan

Principe général et concept

Spécificités Arduino

En pratique

Types d'interruptions

- ▶ Externes
 - ▶ Connexion matérielle
 - ▶ INT0 (pin 2) et INT1 (pin 3)
 - ▶ Détection sur front montant, descendant, changement d'état, signal bas (l'interruption se relance en boucle tant que le signal est à bas).
 - ▶ Changement de broche
 - ▶ interprétation principalement logicielle
 - ▶ un évènement sur une des 20 E/S de l'Arduino déclenche une interruption
 - ▶ recherche dans la routine de la broche qui a déclenché l'interruption
- Plus pratique d'utiliser les interruptions externes

Fronts et signaux

- ▶ front montant



- ▶ front descendant



- ▶ Signal bas



- ▶ Changement d'état



Interruptions externes – Initialisation

- ▶ Étapes d'initialisation d'une interruption
 1. Réglage de l'interruption (registre EICRA)
 2. Activation de l'interruption (registre EIMSK)
 3. Activation de toutes les interruptions (registre SREG)

Interruptions externes – Configuration bas niveau

- ▶ Registre EICRA (External Interrupt Control Register A)

7	6	5	4	3	2	1	0
–	–	–	–	ISC11	ISC10	ISC01	ISC00

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request
0	1	Any logical change on INT0 generates an interrupt request
1	0	The falling edge of INT0 generates an interrupt request
1	1	The rising edge of INT0 generates an interrupt request

Interruptions externes – Configuration bas niveau

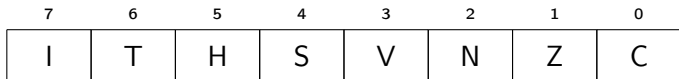
- ▶ Registre EIMSK (External Interrupt Mask Register)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	INT1	INT0

- ▶ INT1 activation des interruptions externes (1 activé, 0 désactivé)
- ▶ INT0 activation des interruptions externes (1 activé, 0 désactivé)

Interruptions externes – Configuration bas niveau

- ▶ Registre SREG (Status Register)
- ▶ Activation globale des interruptions



- ▶ I (Global interrupt Enable) : 1 pour activer les interruptions

Configuration bas niveau

- ▶ Registre EIFR (External Interrupt Flag Register)
- ▶ Flag mis à 1 lorsque l'interruption est déclenchée.

7	6	5	4	3	2	1	0
—	—	—	—	—	—	INTF1	INTF0

- ▶ INTF1 passe à 1 lorsqu'un changement sur la broche a déclenché une interruption
- ▶ INTF0 passe à 1 lorsqu'un changement sur la broche a déclenché une interruption

Temps de traitement

- ▶ Temps de réaction du microcontrôleur aux interruptions
 - ▶ 4 cycles d'horloge pour aller dans la routine
 - ▶ À 16Mhz : 250ns
 - ▶ Même temps pour en sortir
- ▶ Au retour d'une interruption, au moins 1 instruction du programme principal est exécutée avant de repartir dans une interruption en attente

Plan

Principe général et concept

Spécificités Arduino

En pratique

Fonctions bas niveau

```
int pin = 13;
// variable globale pour partager les donnees
volatile int state = LOW;

void setup(void) {
    pinMode(pin, OUTPUT);
    sei(); // Activation globale
    EIMSK |= (1 << INTO); // Activation INTO
    EICRA |= (1 << ISC01); // INTO falling edge
}

void loop(void) {
    // la LED reflète l'état de la variable
    digitalWrite(pin, state);
}

// la fonction attachée à l'interruption INTO
ISR(EXT_INT0_vect) {
    state = !state; // inverse l'état de la variable
}
```

Fonctions haut niveau

```
int pin = 13;
// variable globale pour partager les donnees
volatile int state = LOW;

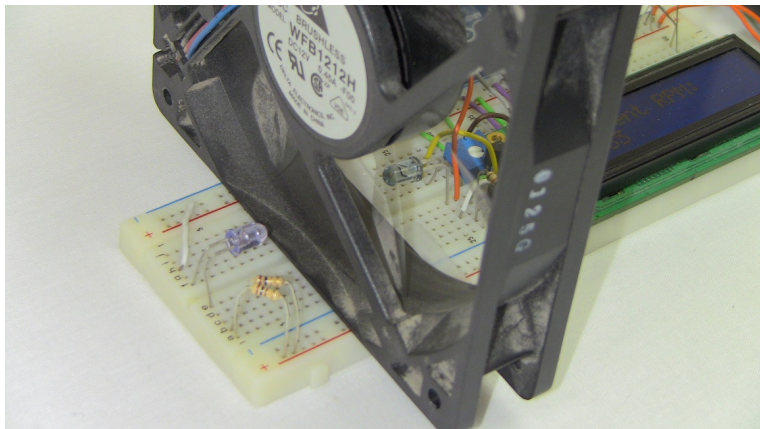
void setup() {
  pinMode(pin, OUTPUT);
  // attache INTO a la fonction blink
  attachInterrupt(0, blink, FALLING);
  // LOW, CHANGE, RISING, FALLING
}

void loop() {
  // la LED reflète l'état de la variable
  digitalWrite(pin, state);
}

// la fonction attachée a l'interruption INTO
void blink() {
  state = !state; // inverse l'état de la variable
}
```

Utilisations – Montages

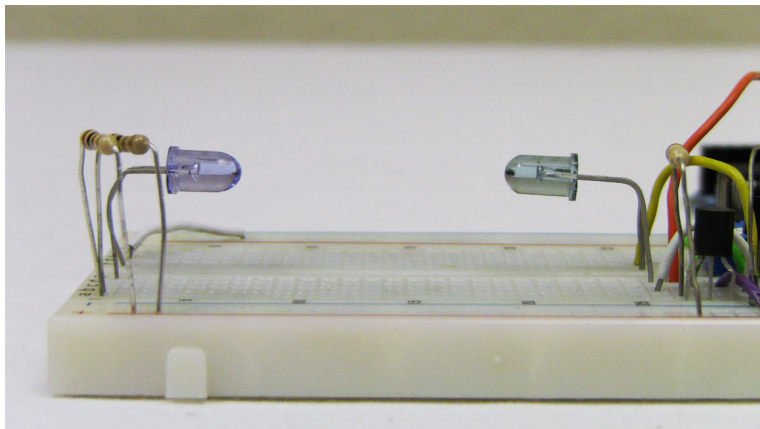
- ▶ Montage possible avec les interruptions



[http ://armando2k.com/](http://armando2k.com/)

Utilisations – Montages

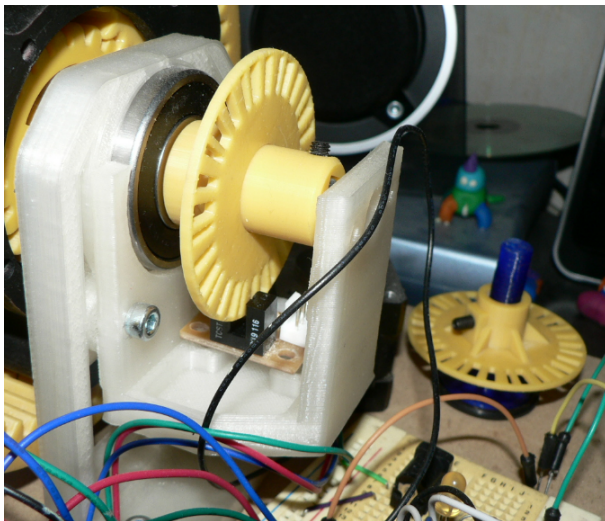
- ▶ Montage possible avec les interruptions



[http ://armando2k.com/](http://armando2k.com/)

Utilisations – Montages

- ▶ Montage possible avec les Interruptions



Utilisations – Montages

- ▶ Montage possible avec les interruptions

