arduino, reset, em

arduino software auto-reset

Comment déclencher un auto-reset de l'arduino ?

Il existe plusieurs moyens pour déclencher un redémarrage / remise à zéro d'une carte arduino en hardware, mais on peut aussi le déclencher en software.

En hardware

- utiliser le bouton reset sur la carte !
- couper l'alimentation... même effet en débranchant / rebranchant le cable USB
- mettre la broche reset à la masse en utilisant un bouton externe, à l'extérieur d'un boîtier par exemple
- déclencher par un circuit externe complémentaire (avec un 555 par exemple)
- relier une broche de l'arduino à la broche reset (déconseillé)

Pour le dernier cas, c'est déconseillé car «*the problem with trying to use a digital output pin to generate a reset pulse is that the when the AVR first starts the reset sequence the first thing it does is set all the I/O pins to tri-state inputs, thus removing the reset pulse too soon such that a full reset does not happen.*»
https://forum.arduino.cc/index.php?topic=42436.msg307346#msg307346

Que fait exactement un reset de la carte arduino ?

- replacer les registres I/O de l'ATmega328P à leur valeur initiale
- démarrer le bootloader qui attend quelques secondes un éventuel signal série pour charger un nouveau programme (la led clignote)
- puis démarrer le micrologiciel qui est en mémoire

Tous les détails sur la procédure de *reset* sont dans la datasheet de l'ATmega328P («*system control and reset*», page 40 sur 294).

On y retrouve les 4 causes qui peuvent produire un *reset* :

- Power-on reset. The MCU is reset when the supply voltage is below the power-on reset threshold (V POT).
- External reset. The MCU is reset when a low level is present on the RESET pin for longer than the minimum pulse length.
- Watchdog system reset. The MCU is reset when the watchdog timer period expires and the watchdog system reset mode is enabled.
- Brown-out reset. The MCU is reset

Que se passe t'il quand on envoie un programme ?

Un signal déclenche la séquence de *reset*, puis un signal prévient le *bootloader* qu'il va recevoir des données pour programmer la mémoire flash, une fois chargé le programme démarre.

Et en software alors ?

Deux méthodes qu'on peut trouver sur le web, mais qui **ne produisent pas un vrai reset**, elles permettent plutôt de reprendre le programme à zéro (sans réinitialiser les registres ou recharger le bootloader)

```
/* arduino software reset / test 1  */

void(* resetFunc) (void) = 0; //declare reset function @ address 0

void setup() {
}

void loop() {
  if (millis() > 10000) resetFunc();  //call reset
```

```
}
```

Pour comprendre la syntaxe `void(* resetFunc) (void) = 0;` voir :
http://jdurrett.ba.ttu.edu/3345/handouts/RL-rule.html

«It defines a pointer to a function, and sets that pointer to zero, so when you call the function, you transfer control to the reset vector at address zero.» https://forum.arduino.cc/index.php?topic=385427.msg2656332#msg2656332

«A jump to program location 0 is not a true reset, it just starts your program from the start. It does not reset the internal functions nor starts the auto bootloader.» https://forum.arduino.cc/index.php?topic=42436.msg307346#msg307346

«Option is a null pointer assignment of a function pointer that actually points to nothing when calling the function. Calling this is illegal, cause a null pointer exception and without exception handling results in a reboot. That is a dirty trick and unreliable to recommend. For all believers, the pointer is a pointer to a VOLATILE memory location, not a program memory (flash) location.»

Avec le même principe, on peut utiliser une instruction en assembleur qui point le programme sur l'adresse 0, mais là non plus pas de remise à zéro des registres ni redémarrage du bootloader.

```
void setup() {
}

void loop() {
  if (millis() > 10000) softReset();  //call reset
}

void softReset(){
  asm volatile ("  jmp 0");
}
```

La bonne méthode!

Enfin la bonne méthode consiste à déclencher le watchdog, qui déclenchera lui-même la séquence de reset (mais attention, cette méthode peut provoquer des boucles de resset sans fin sur des bootloaders anciens, dont le WDT n'est pas activé et certaines cartes arduino»)

```
/* Arduino software reboot

   Principe :
   Le watchdog est déclenché, ne recevant pas le signal approprié, il lance la procédure de reboot,
   de la même manière que si on appuyait sur le bouton de la carte.
   Il existe d'autres méthodes, mais celle-ci est la seule qui fasse un reset complet.

   Avertissements :
     * cette méthode n'est pas compatible avec les anciennes versions de bootloader qui n'ont pas le WDT activé (WatchDog Timer)
     * cette méthode peut poser problème sur des clones nano (toujours pour des questions de bootloader)
       cf. https://forum.arduino.cc/index.php?topic=461948.0

   Testé (ok!) sur un Uno R3-DIP de Joy-it

   Sources : multiples discussions sur le forum arduino et ailleurs, par exemple
     * https://www.codeproject.com/articles/1012319/arduino-software-reset
  */

#include <avr/wdt.h>

void setup() {
  Serial.begin(9600);
  delay(1000);
  Serial.println("boot");
}

void loop() {
  if (millis() > 10000) reboot();
}

void reboot() {
  wdt_enable(WDTO_15MS); // activer le watchdog
  while (1) {};          // et attendre ...
 }
```

De cette manière on peut également **déclencher un reset en envoyant un signal sur le port série** (mêmes réserves que la précédente) :

```
/* Arduino software reboot à la réception d'un message série  */

#include <avr/wdt.h>

void setup() {
  Serial.begin(9600);
```

```
  delay(1000);
  Serial.println("boot");
}

void loop() {

  if (Serial.available() > 0) {
    char ch = Serial.read();
    if (ch == '0') {
      Serial.println("arduino va redémarrer");
      delay(1000);
      reboot();
    }
  }
}

void reboot() {
  wdt_enable(WDTO_15MS); // activer le watchdog
  while (1) {};          // et attendre ...
}
```

A signaler également, deux *libraries* qui permettent d'appliquer ces méthodes.

- https://github.com/adafruit/Adafruit_SleepyDog
- https://github.com/qub1750ul/Arduino_SoftwareReset

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/ressource/electronique/arduino/software_auto_reset**
Article mis à jour: **2021/02/16 23:57**