

Curso avanzado sobre Arduino: Interrupciones Hardware

ElCacharreo.com



ElCacharreo.com



Introducción a Arduino: Presente



Introducción a Arduino: Presente



José Antonio Vacas Martínez

blog
javacasm@elcacharreo.com
twitter
linkedin



Arduino avanzado: Interrupciones

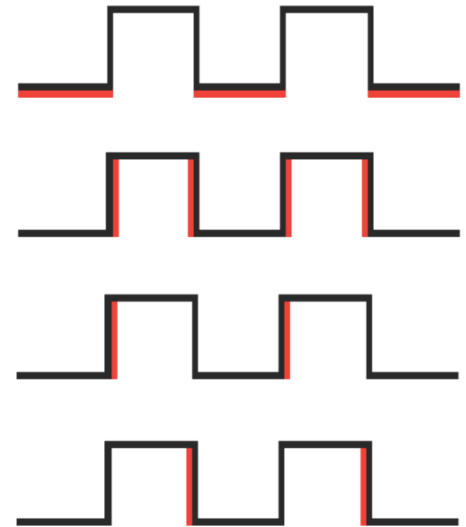
Interrupciones hardware:

Las interrupciones externas llamadas INT0 e INT1 están asociadas a los pines 2 y 3 (en arduino UNO, el mega tiene otras 4: 2 (pin 21), 3 (pin 20), 4 (pin 19) y 5 (pin 18))

Podemos configurarlas dependiendo del cambio de estado como

de subida ->	RISING
de bajada ->	FALLING
cambio ->	CHANGE
estado bajo ->	LOW

Son muy rápidas.



Arduino avanzado: Interrupciones

```
int pin = 13;
volatile int estado = LOW;

void setup()
{
    pinMode(pin, OUTPUT);
    attachInterrupt(0, parpadeo, CHANGE);
}

void loop()
{
    digitalWrite(pin, estado);
}

void parpadeo()
{
    estado = !estado;
}
```



Arduino avanzado: Interrupciones

Podríamos estar tentados de mover la llamada de `digitalWrite` al método `parpadeo`. Lo hemos dejado así por 2 razones:

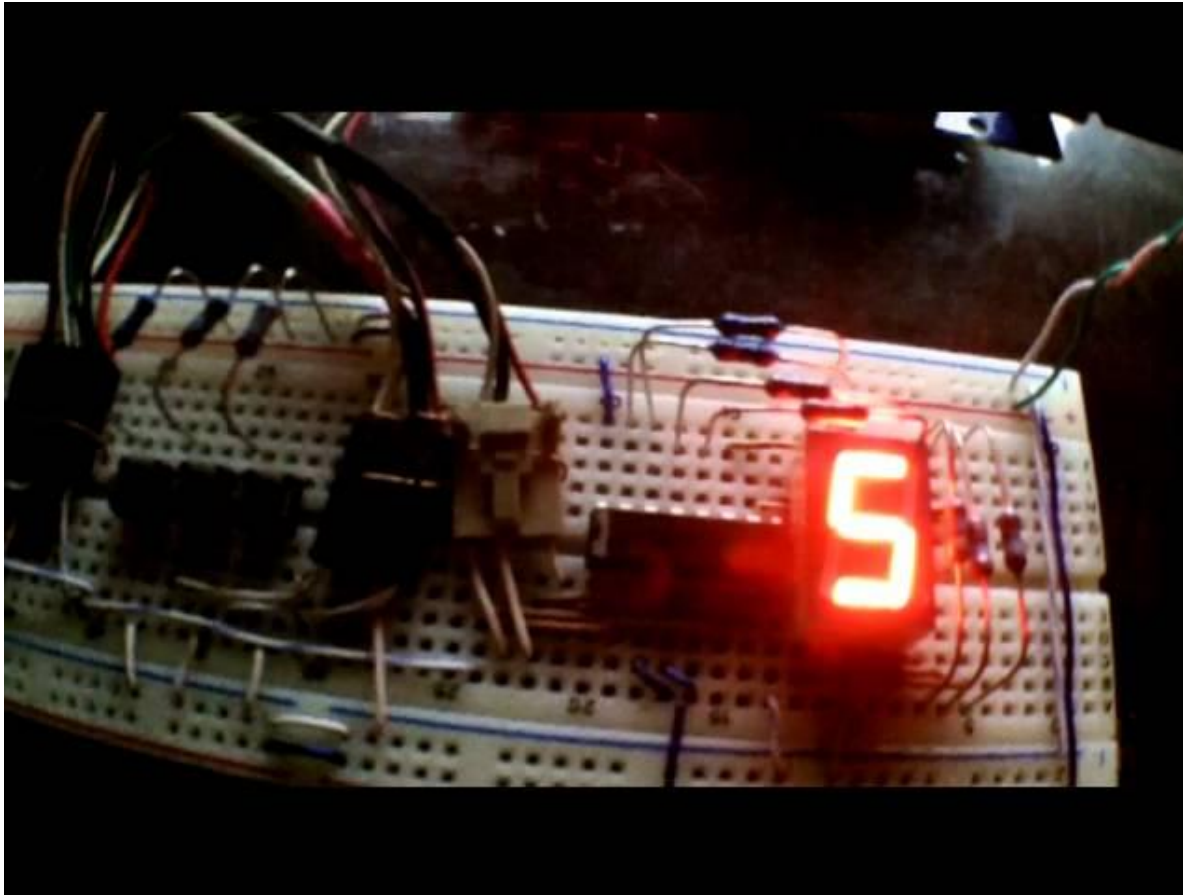
- El código ejecutado en las funciones de las interrupciones debe ser lo más corto posible. No olvidemos que estamos interrumpiendo el código "central"
- Así demostramos que estamos haciendo parte del procesamiento en 2 sitios distintos

Hay que tener cuidado con los cambios que se hacen a las variables en una interrupción pues podrían entrar en conflicto con los valores que tenían, por eso se marcan esas variables como "volatile" para el compilador lo tenga en cuenta.

Los micro ATmega tienen muchas más posibilidades de interrupciones, pero no todas están soportadas por arduino.



Interrupciones Hardware



Introducción a Arduino: Proyectos

```
const int BTN1 = 18; // Botones
const int BTN2 = 19;
const int BTN3 = 20;
// Display 7-Segmentos
const int D0 = 4;
const int D1 = 5;
const int D2 = 6;
const int D3 = 7;
volatile int inc; // Contador
volatile int onOff; // On/off
volatile long lastInt; // Tiempo last interrupción.
// Incrementa y escribe el número al display
void increment() {
    inc++;
    inc = (inc < 10) ? inc : 0;
    writeTo7Seg(inc); }
// Decrementa y escribe el número al display
void decrement() {
    inc--;
    inc = (inc >= 0) ? inc : 9;
    writeTo7Seg(inc);}
```

```
void setup() {
    pinMode(D0, OUTPUT);
    pinMode(D1, OUTPUT);
    pinMode(D2, OUTPUT);
    pinMode(D3, OUTPUT);
    inc = 0;
    lastInt = 0;
    onOff = 1; // Display encendido al encender
    // Asociamos la interrupción a nuestro código
    attachInterrupt(5, int5, RISING); // línea 18
    attachInterrupt(4, int4, RISING); // línea 19
    attachInterrupt(3, int3, RISING); // línea 20
    // Colocamos un 0 en nuestro display de 7 segmentos
    writeTo7Seg(inc); }
void int5() { increment(); }
void int4() { decrement(); }
void int3() {
    onOff = onOff ^ 1;
    if (onOff) writeTo7Seg(15);
    else writeTo7Seg(inc);
}
```



Conclusiones

Gracias por vuestra atención

