

Interrupciones

Las interrupciones detienen la ejecución del programa del Arduino para que otra tarea pueda ser ejecutada.

El programa principal se ejecuta de manera normal pero, cuando ocurre una interrupción, el programa principal se detiene mientras la otra rutina es ejecutada. Cuando esta rutina termina, el procesador regresa a la ejecución del programa principal.

Características

- Las interrupciones pueden venir de varias partes, ya sea hardware o software.
- La mayoría de las placas Arduino tienen dos interrupciones de hardware, llamadas interrupt0 e interrupt1, con conexión física a los pines digitales 2 y 3, respectivamente.
- El Arduino Mega tiene seis interrupciones de hardware incluyendo interrupciones adicionales, interrupt2 hasta interrupt5, en los pines 21, 20, 19 y 18.
- Se puede definir una rutina utilizando una función especial llamada "Interrupt Service Routine" o ISR.

Es posible que dicha función se ejecute de manera automática, cada vez que un evento se detecte un evento en un pin.

Tipos de Interrupciones

Existen dos tipos de interrupciones:

- Interrupciones de Hardware - Ocurren en respuesta a un evento externo, como el cambio de estado de un pin.
- Interrupciones de Software - Ocurren en respuesta a una instrucción enviada por software, utilizando la función attachInterrupt().

Generalmente, una interrupción debe ser lo más corta y rápida posible. Si un programa utiliza varias interrupciones, solo se ejecuta una a la vez. Las otras interrupciones se ejecutarán en el orden de prioridad que tengan hasta que la interrupción actual en ejecución termine.

Típicamente, se usan variables globales para pasar datos entre interrupciones y el programa principal. Dichas variables deben ser declaradas como volatile para asegurar que los datos de la variable se escriban directamente en memoria y no en los registros del micro-controlador.

Sintaxis

`attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);`

- `pin`: número de pin
- `ISR`: El nombre de la función que se llamará cuando ocurra la interrupción. Esta función no debe tomar parámetros ni regresar ningún valor.
- `mode`: Define cuando debe ejecutarse la interrupción. Existen cuatro valores predefinidos:
 - `LOW` - ejecuta la interrupción cuando el pin está en estado bajo o `LOW`
 - `CHANGE` - ejecuta la interrupción cuando el pin cambia de estado
 - `RISING` - ejecuta la interrupción cuando el pin sube de `LOW` a `HIGH`
 - `FALLING` - ejecuta la interrupción cuando el pin baja de `HIGH` a `LOW`
 - `HIGH` - ejecuta la interrupción cuando el pin está en estado `HIGH`

Ejemplo

```
const byte ledPin = 13;
const byte interruptPin = 2;
volatile byte state = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(interruptPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink,
CHANGE);
}

void loop() {
  digitalWrite(ledPin, state);
}

void blink() {
  state = !state;
}
```

Referencias:

Arduino Language Reference (2018). Recuperado de:

<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>