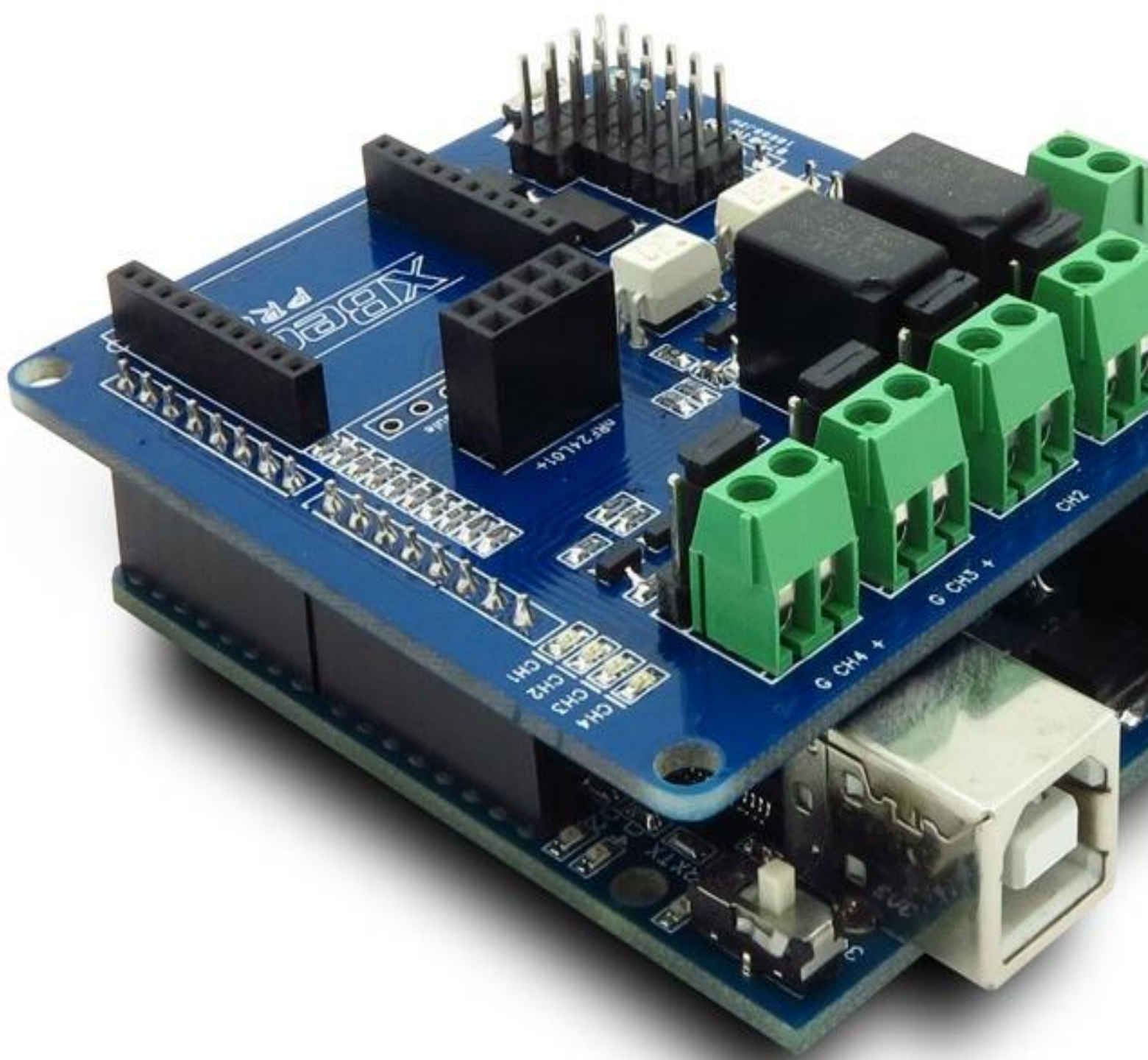
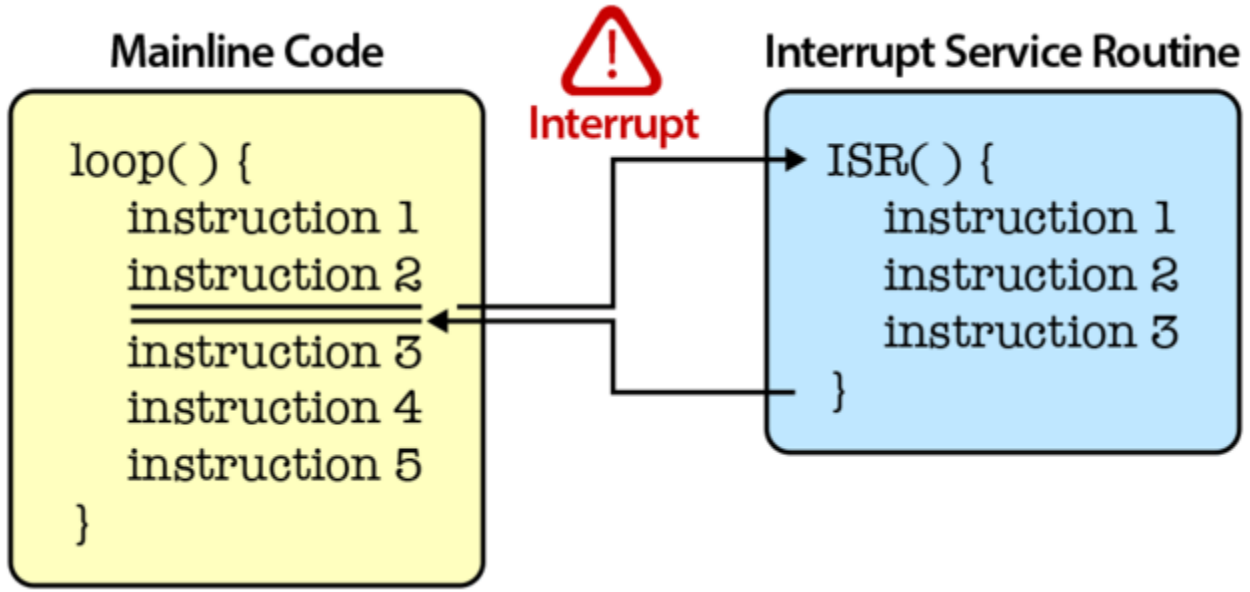
INTERRUPCIONES

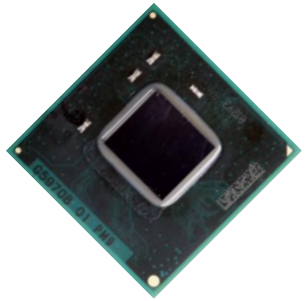
SISTEMAS EMBEBIDOS



Interrupciones

Partes especiales de Arduino para poder trabajar por fases el programa

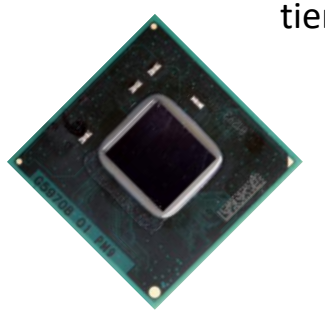




Interrupciones Según el Arduino

Tablero	int.0	Int.1	Int.2	int.3	int.4	int.5
Uno, Ethernet	2	3				
Mega2560	2	3	21	20	19	18
Leonardo	3	2	0	1	7	

Las interrupciones son útiles para hacer que las cosas sucedan de forma automática en los programas de microcontroladores, y pueden ayudar a resolver problemas de tiempo.



attachInterrupt () es la función que no permite trabajar con interrupciones, es necesario la creación de subrutinas

AttachInterrupt()

attachInterrupt(interrupt,ISR,modo)

Interrupt: el número de la interrupción
ISR: Subrutina a llamar
Modo: Definir el modo de trabajo
- LOW LEVEL
- CHANGE
- RISING
- FALLING

Sintaxis:

attachInterrupt(0,INT0,CHANGE)

Subrutina:
Sintaxis:

```
void INT0 ( )
{
  staado=1-estado ;
}
```

Código
Attach.ino

```
#define led 10
volatile int state=LOW;
int on=0;
void setup() {
 attachInterrupt (0,INT,LOW);
 pinMode(led,OUTPUT);
}

void loop() {
if(on==1){
  digitalWrite ( led , state = ! state) ;
  delay(500);
  }
}
void INT()
{
on=1-on;
}
```

Configuración de Interrupción

Estructura de toogle

Subrutina de interrupción para activar al led.

Código
SerialEvent.ino

```
#define led 10
volatile int state=LOW;
int tiempo=500;
char inChar;

void loop() {
  digitalWrite ( led , state = ! state) ;
  delay(tiempo);
}

void serialEvent(){
  while (Serial.available()) {
  inChar=(char)Serial.read();
    if(inChar=='A'){
      tiempo=100;
    }
   }
}
```

Programación de toogle del led, con delay manejado por interrupciones

Subrutina de interrupción de comunicación serial

Estructura de recepción de dato mediante interrupción

PROGAMACIÓN

CÓDIGO 16: MANEJO DE INTERRUPCIONES