UNIVERSIDAD DE ANTIOQUIA

1 8 0 3

Facultad de Ciencias Exactas y Naturales

Introducción a las Tecnologías 4.0

Módulo III - IoT

Daniel Estrada

Físico

Grupo de Instrumentación Científica y microelectrónica

Medellín, Mayo 2023

Contenido

- Arduino - Característica avanzadas
- Transmisión de los datos
- Actividad práctica 3
- Primera aproximación a las bases de datos
- Actividad práctica 4

Arduino - características avanzadas

Motivación

A veces es necesario ir a una programación de más bajo nivel para tener un mayor control y optimización de los dispositivos conectados

Registros

Si cada ciclo de reloj es importante no se pueden usar las funciones ordinarias de Arduino. (Generar señales, Monitorear varios sensores).

Control de pines

Interrupciones

digitalWrite()

```
138 void digitalWrite(uint8_t pin, uint8_t val)
139 {
140     uint8_t timer = digitalPinToTimer(pin);
141     uint8_t bit = digitalPinToBitMask(pin);
142     uint8_t port = digitalPinToPort(pin);
143     volatile uint8_t *out;
144
145     if (port == NOT_A_PIN) return;
146
147     // If the pin that support PWM output, we need to turn it off
148     // before doing a digital write.
149     if (timer != NOT_ON_TIMER) turnOffPWM(timer);
150
151     out = portOutputRegister(port);
152
153     uint8_t oldSREG = SREG;
154     cli();
155
156     if (val == LOW) {
157         *out &= ~bit;
158     } else {
159         *out |= bit;
160     }
161
162     SREG = oldSREG;
163 }
```

delay()

```
106 void delay(unsigned long ms)
107 {
108     uint32_t start = micros();
109
110     while (ms > 0) {
111         yield();
112         while ( ms > 0 && (micros() - start) >= 1000) {
113             ms--;
114             start += 1000;
115         }
116     }
117 }
```

Arduino - características avanzadas

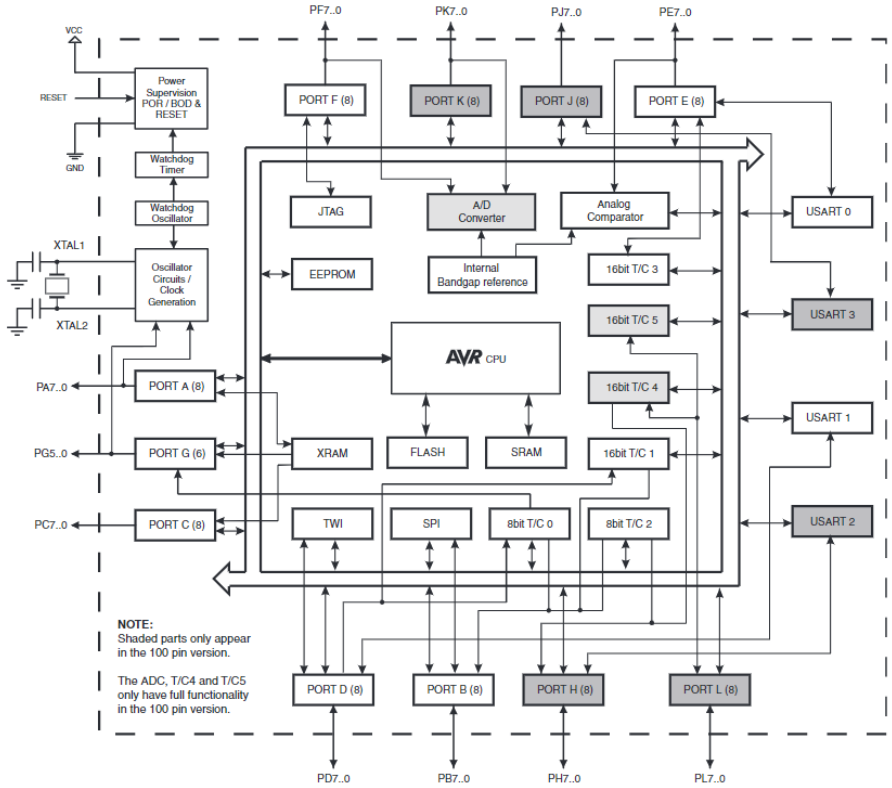
Motivación

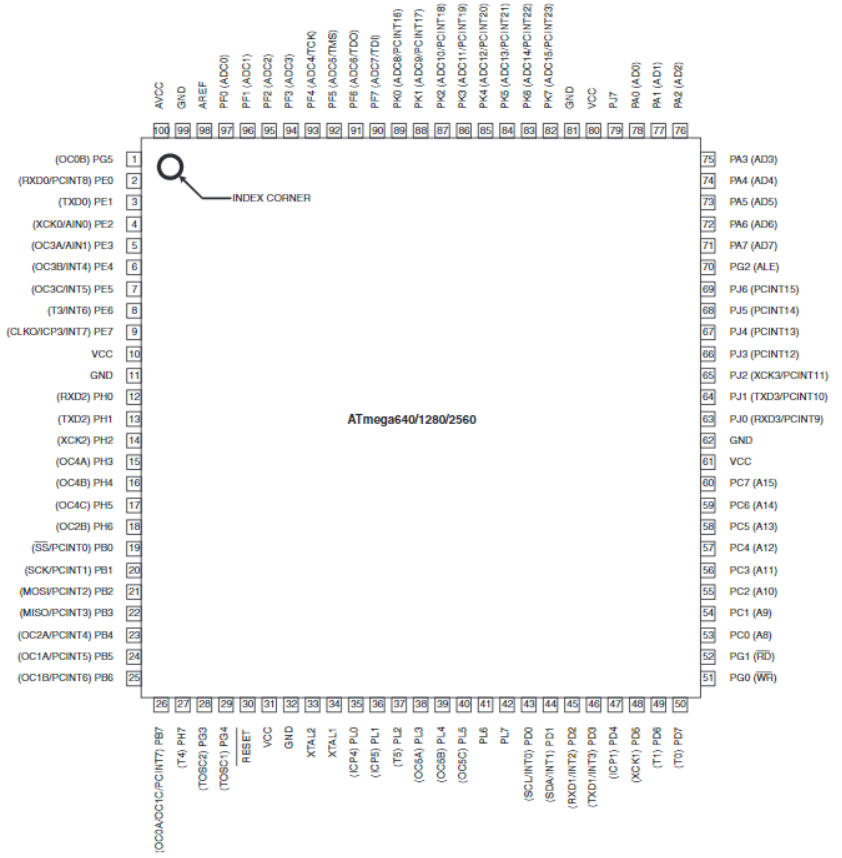
Registros

Control de pines

Interrupciones

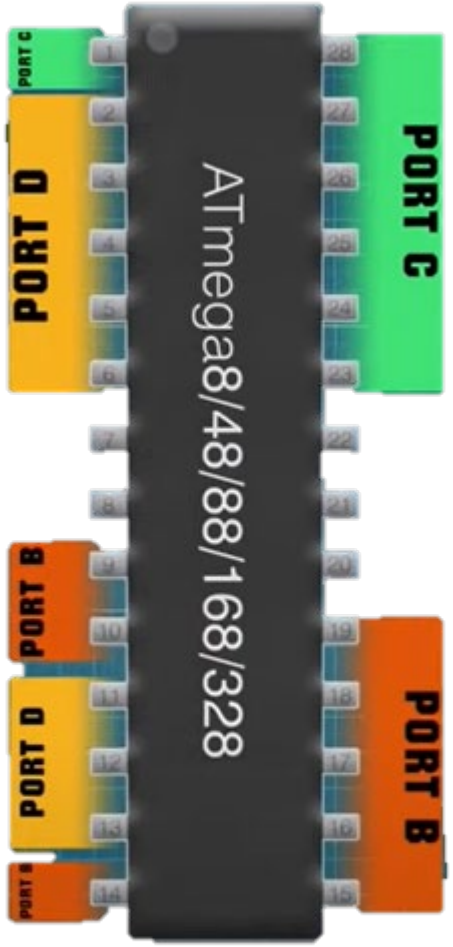
Cada aspecto del microcontrolador se maneja a través de sus registros internos. Lo que hacen las funciones de alto nivel es modificarlos según lo deseado.





Arduino - características avanzadas

- Motivación
- Registros
- Control de pines
- Interrupciones



Los pines de I/O se agrupan por puertos (8 pines) etiquetados A, B, … Se pueden manipular a través de los registros asociados a cada puerto.

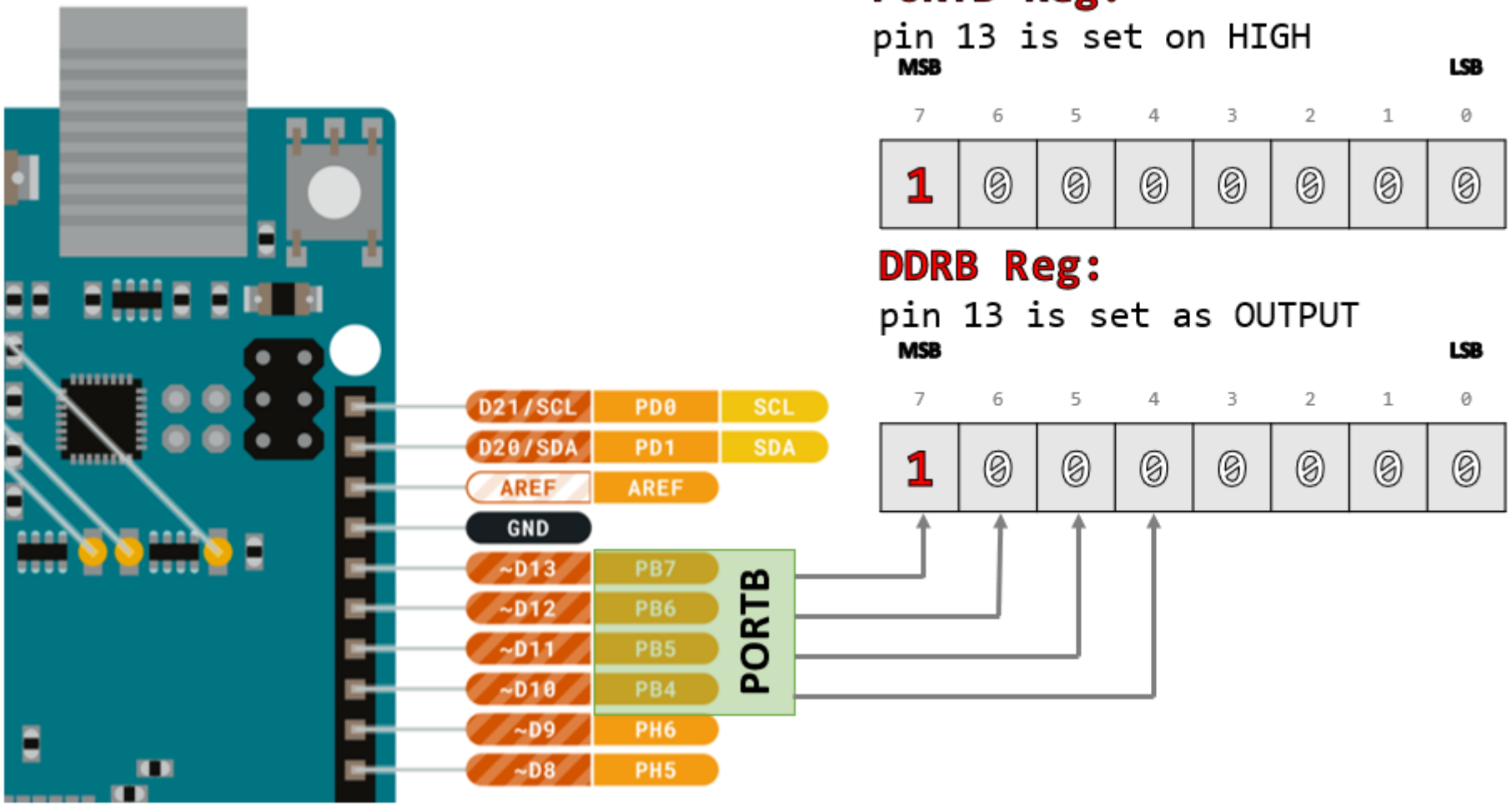
DDR# (registro de dirección de datos)
PORT# (registro de salida)
PIN# (registro de entrada)

Arduino - características avanzadas

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ciencias Exactas y Naturales

- Motivación
- Registros
- Control de pines
- Interrupciones

Ejemplo: PIN 13



Ejemplo: PIN 13

```
1 void setup(){
2     pinMode(9, OUTPUT); //
3 }
4 void loop(){
5     digitalWrite(9, HIGH);
6     digitalWrite(9, LOW);
7     delay(5);
8 }
```

```
1 void setup(){
2     DDRB |= B10000000; //
3 }
4 void loop(){
5     PORTB |= B10000000; /
6     PORTB &= !B10000000;
7     delay(5);
8 }
```

Arduino - características avanzadas

Motivación
Registros
Control de pines
Interrupciones

Manipular los registros evitando manipular los demás es importante. Hay que usar operadores booleanos.

Operator	Function
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR (Exclusive OR)
<<	LEFT SHIFT
>>	RIGHT SHIFT

Si se quiere poner en “1”, OR:

`0101|0010 → 0111`

Si se quiere poner en “0”, AND:

`0111&1101 → 0101`

`= 0111&not(0010)`

Arduino - características avanzadas

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ciencias Exactas y Naturales

Motivación

Registros

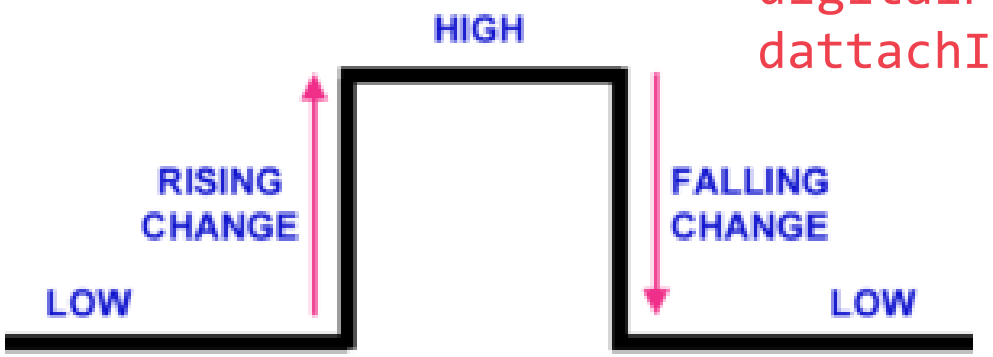
Control de pines

Interrupciones

Para responder de manera rápida, se usan señales que ocasiona una **suspensión temporal de la ejecución del programa principal y** se ejecuta una subrutina de servicio de interrupción (ISR - Interruption Service Rutine).

Se distinguen al menos dos EXTERNAS y PROGRAMADAS (de TIMER).

```
attachInterrupt(#interruptPin, ISR, mode)
digitalPinToInterrupt(#pin)
dattachInterrupt(#interruptPin)
```



Solo para pines -> 2, 3, 18, 19, 20, 21.

Ejemplo - Botón

Arduino - características avanzadas

Motivación
Registros
Control de pines
Interrupciones

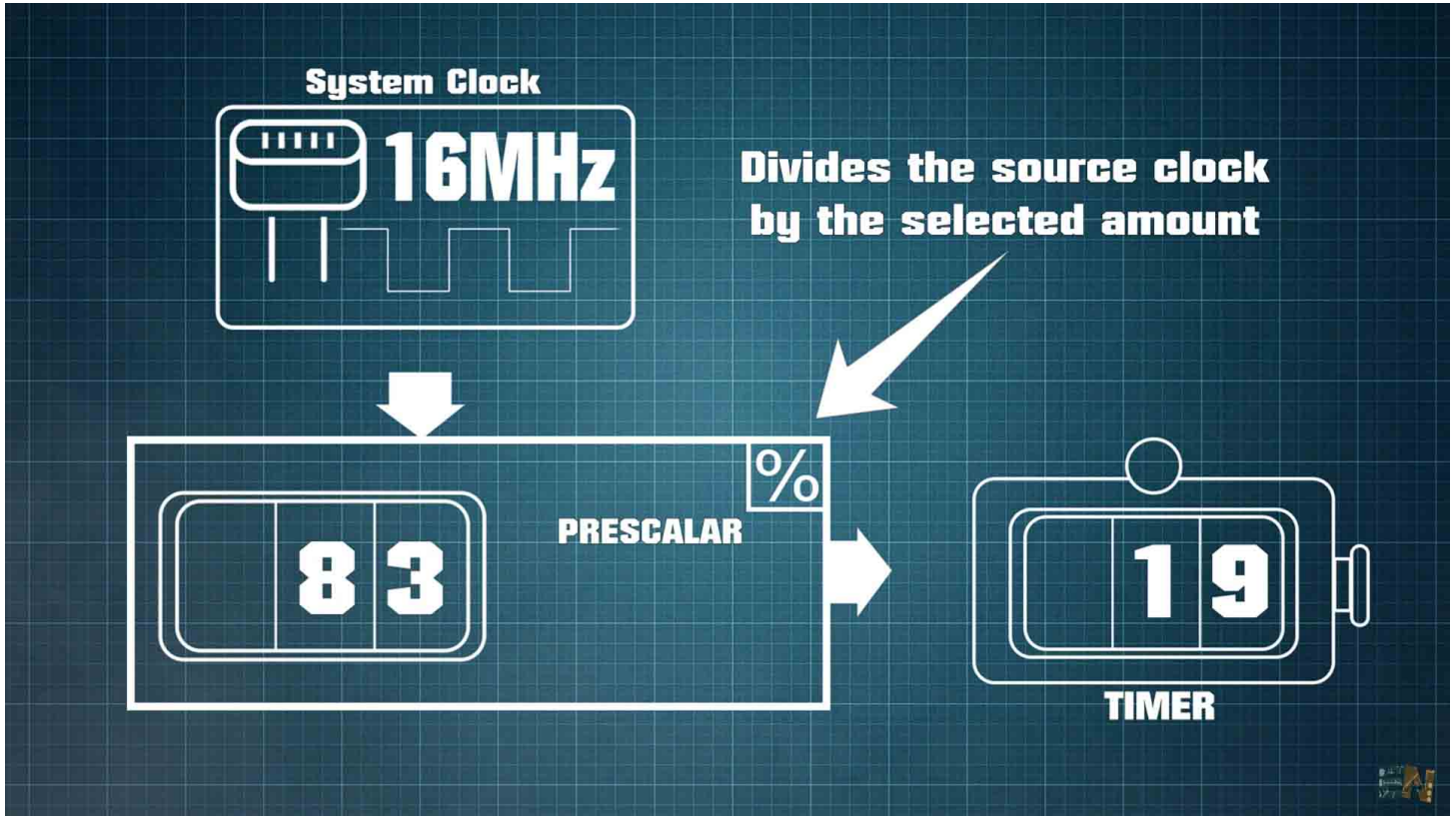
Para responder de manera rápida, se usan señales que ocasiona una **suspensión temporal de la ejecución del programa principal y** se ejecuta una subrutina de servicio de interrupción (ISR - Interruption Service Rutine).

Se distinguen al menos dos EXTERNAS y PROGRAMADAS (de TIMER).

Arduino - características avanzadas

Motivación
Registros
Control de pines
Interrupciones

La configuración se hace a través de los registros de TIMER (0, 1, ..)



Arduino - características avanzadas

UNIVERSIDAD DE ANTIOQUIA
Facultad de Ciencias Exactas y Naturales

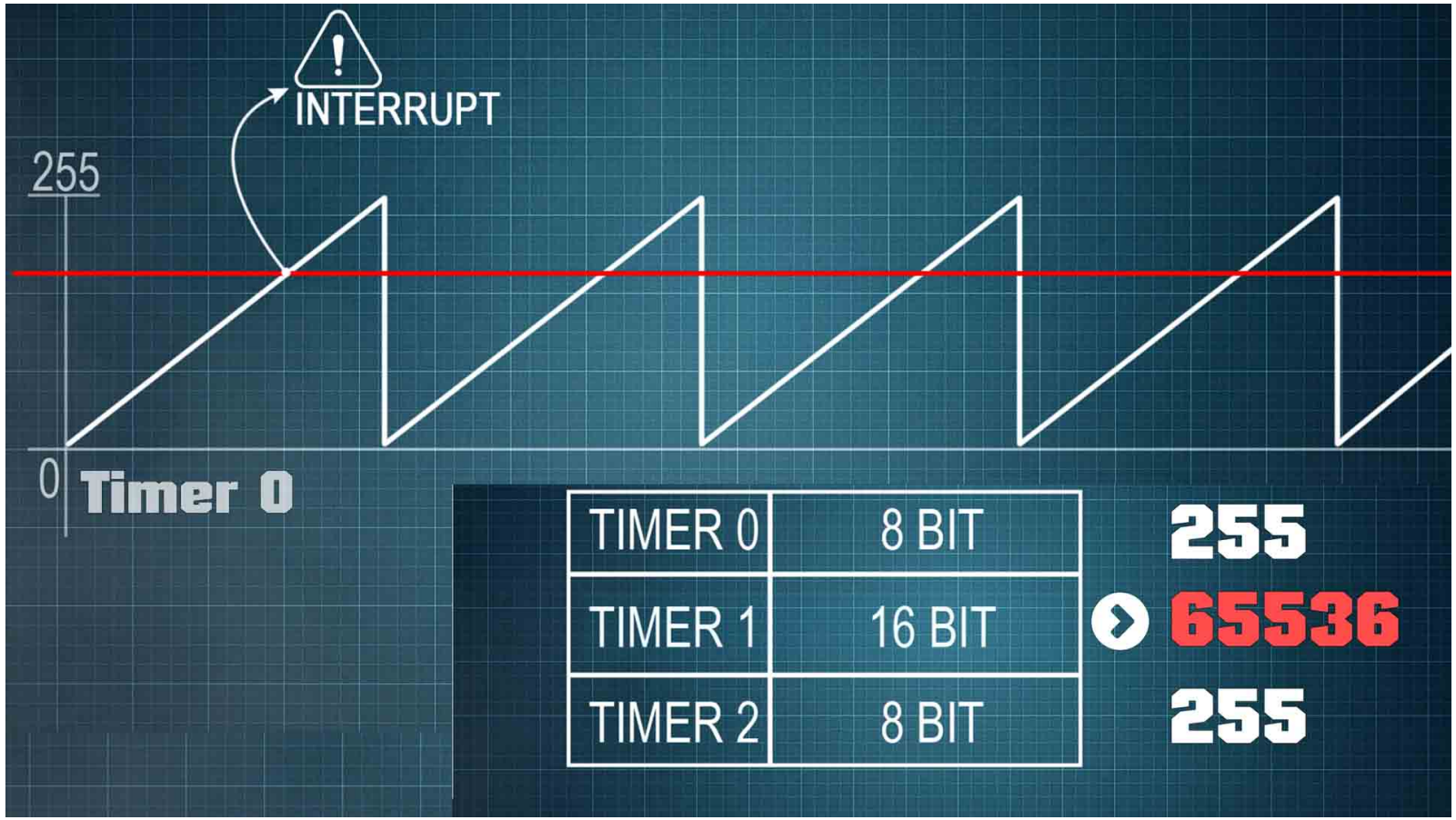
Motivación

Registros

Control de pines

Interrupciones

La configuración se hace a través de los registros de TIMER (0, 1, ..)



Ejemplos - TIMERS

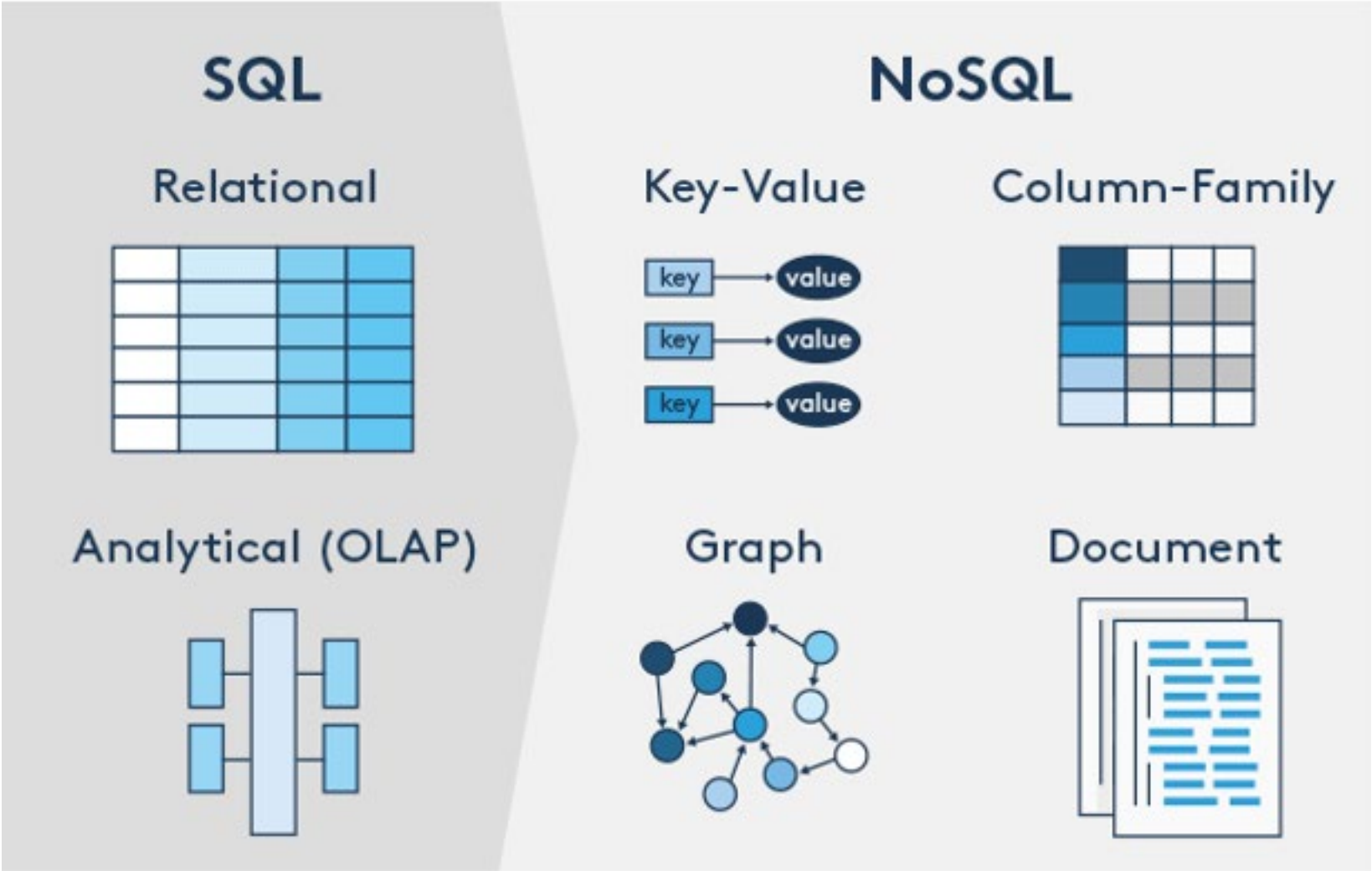
Actividad 3

La idea es llevar los datos por serial controlando las frecuencias de muestreo con los TIMER.

Actividad 4

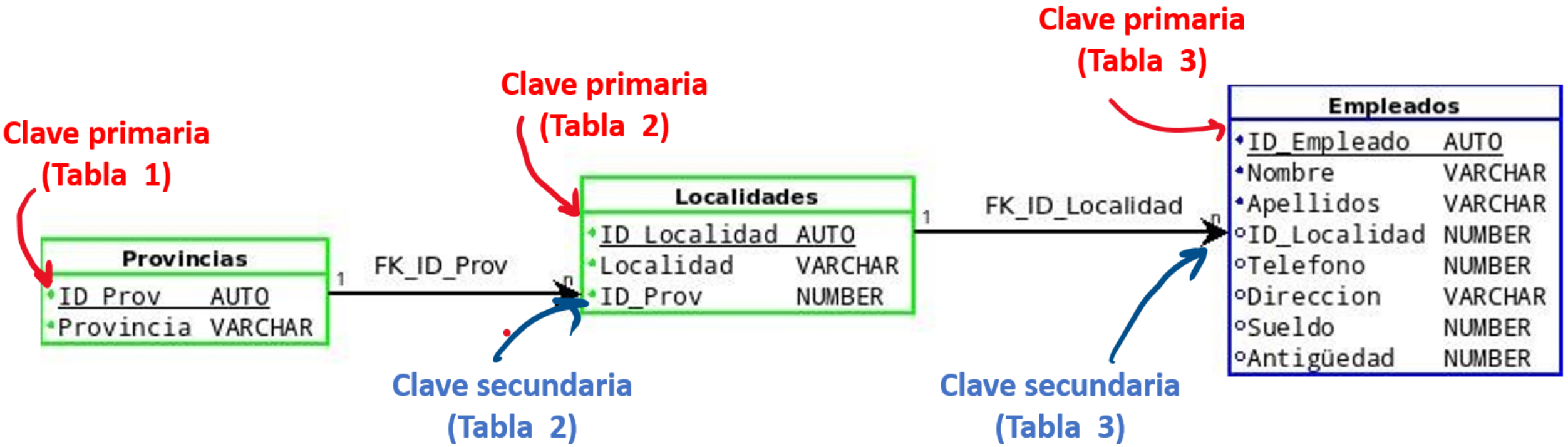
Hay que leer los datos enviados por el Arduino y guardarlos en una base de datos

Base de datos

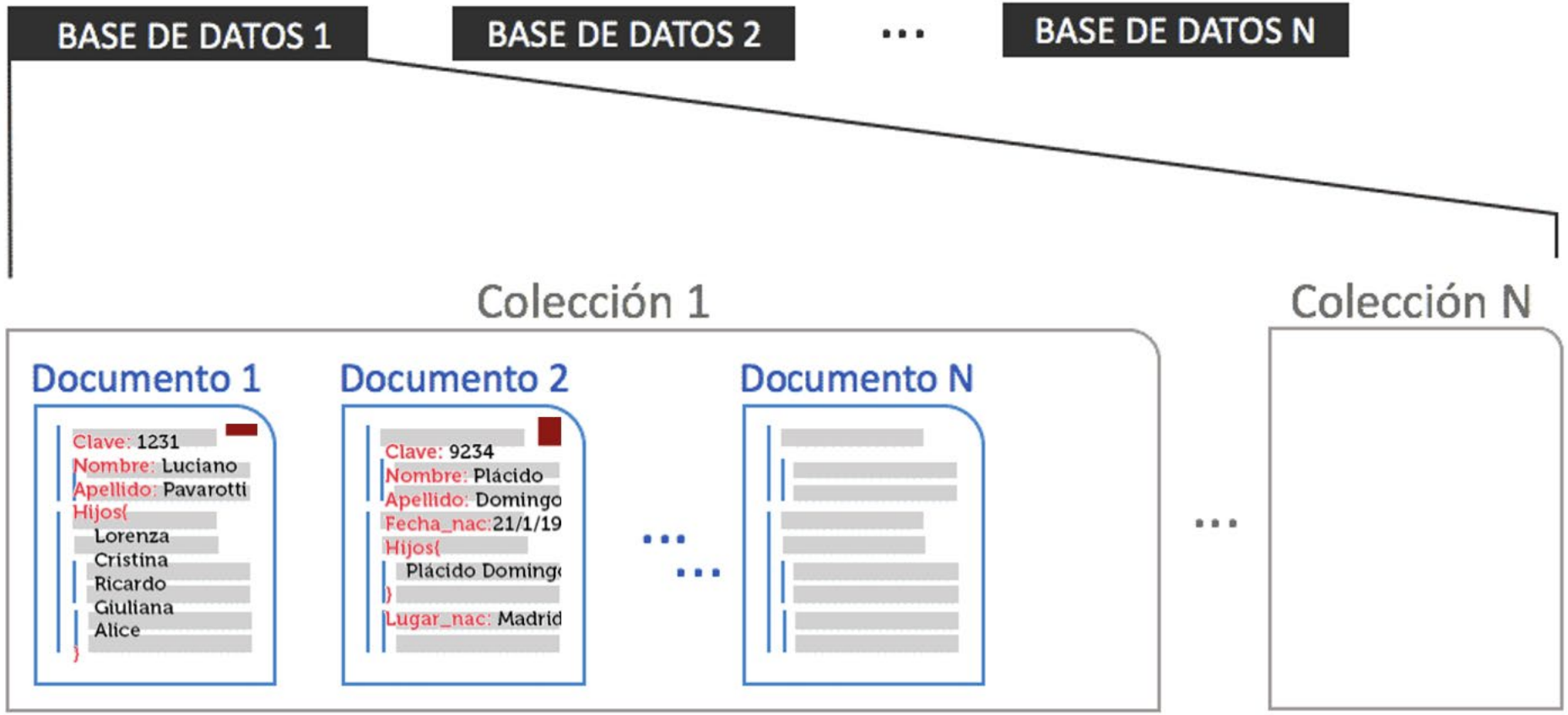


→ Lenguaje de consultas : SQL (Structured Query Language).

Base de datos - relacional



Base de datos – no relacional (documental)



mongoDB®