

## I2C en Arduino

*Departamento de Automática*



Universidad de Alcalá



/gso>

## Biblioteca de funciones Wire

- Biblioteca de funciones externa para controlar un bus I2C
- Pines SDA/SCL pre-programados:
  - Arduino Uno: A4 (SDA), A5 (SCL)
  - Arduino Mega: 20 (SDA), 21 (SCL)
- Utiliza las resistencias de pull-up internas de Arduino
- Implementa direcciones I2C de 7 bits

## Función de inicialización

- Función de inicialización de la comunicación:
  - `Wire.begin(address)`
  - Configura el dispositivo para utilizar el canal I2C empleando los pines por defecto
  - Recibe como parámetro la dirección del dispositivo en el bus
  - En el caso del maestro, el parámetro es opcional

Dispositivo maestro:

```
void setup() {  
  Wire.begin();  
}
```

Dispositivo esclavo con dirección 4:

```
void setup() {  
  Wire.begin(4);  
}
```

## Maestro escritor → esclavo lector

- Dispositivo maestro:
  - Inicio de la transmisión:
    - `Wire.beginTransmission(address)`
  - Envío de datos:
    - `Wire.write(value/string/data, length)`
  - Fin de transmisión:
    - `Wire.endTransmission()`

```
Wire.beginTransmission(4);  
Wire.write(...);  
Wire.write(...);  
...  
Wire.endTransmission();
```

## Maestro escritor → esclavo lector

- Dispositivo esclavo:

- Instalación del manejador
  - `Wire.onReceive(function)`
- Recepción de datos:
  - `Wire.available()`
  - `Wire.read()`

```
void setup() {  
    Wire.begin(4);  
    Wire.onReceive(function)  
}  
  
void function(int numBytes) {  
    while (Wire.available()) {  
        char c = Wire.read();  
        ...  
    }  
}
```

- El manejador es una función que se llama cada vez que se inicia una transmisión
  - El manejador lee los datos con la función `Wire.read()`
  - La función `Wire.available()` devuelve el número de datos pendientes

## Maestro lector → esclavo escritor

- Dispositivo maestro:
  - Petición de datos
    - `Wire.requestFrom(address, quantity)`
  - Recepción de datos:
    - `Wire.available()`
    - `Wire.read()`

```
Wire.requestFrom(4, 10);  
  
while (Wire.available()) {  
    char c = Wire.read();  
    ...  
}
```

## Maestro lector → esclavo escritor

- Dispositivo esclavo:

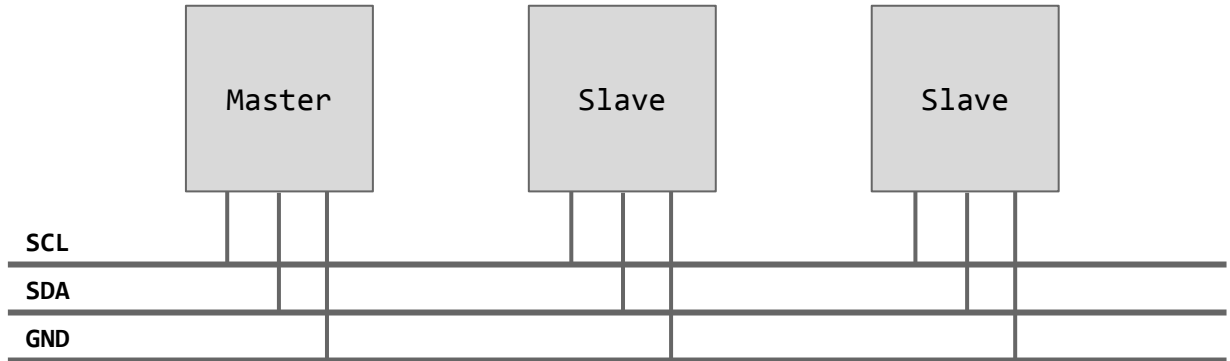
- Instalación del manejador
  - `Wire.onRequest(function)`
- Envío de datos:
  - `Wire.write(value/string/data, length)`

```
void setup() {  
  Wire.begin(4);  
  Wire.onRequest(function)  
}  
  
void function() {  
  Wire.write(...);  
}
```

- El manejador es una función que se llama cada vez que se inicia una transmisión que solicita el envío de datos
  - El manejador envía los datos con la función `Wire.write()`

## Conexión entre múltiples Arduinos

- Ejercicio propuesto: conexión entre un maestro y N esclavos



- El maestro envía un dato (ON/OFF) a los esclavos de forma periódica
- Cuando un esclavo recibe el dato, enciende (ON) o apaga (OFF) el LED\_BUILTIN





© Departamento de Automática. Universidad de Alcalá. Este documento se ha publicado con la licencia Creative Commons Attribution Share-Alike 4.0 (international): <https://creativecommons.org/licenses/by-sa/4.0/>