

Práctica 4: I2C en Arduino



Departamento de Automática
Universidad de Alcalá

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

Maestro Escritor → Esclavo Lector

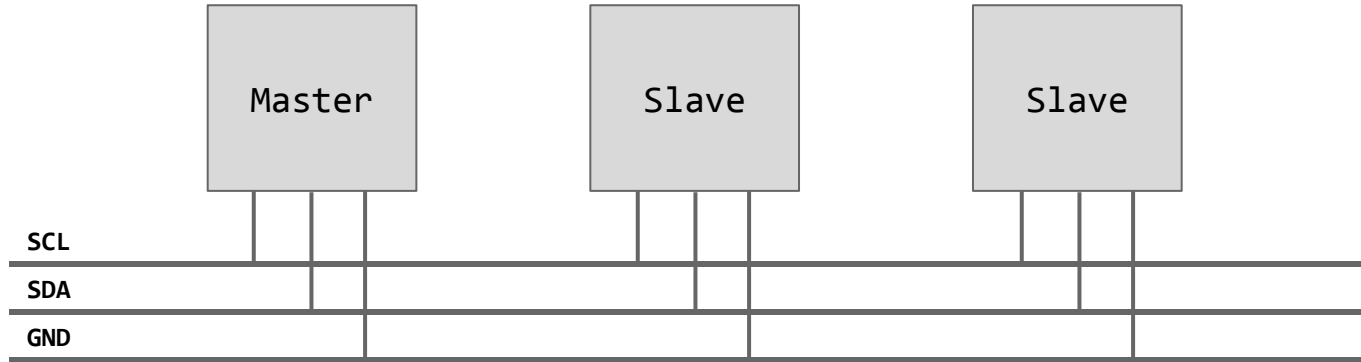
- Función de inicialización de la comunicación
`(Wire.begin(address))`
- Dispositivo Maestro:
 - Inicio de transmisión `(Wire.beginTransmission(address))`
 - Envío de datos `(Wire.write(value/string/data,length))`
 - Fin de transmisión `(Wire.endTransmission())`
- Dispositivo Esclavo:
 - Instalación de manejador `(Wire.onReceive(function))`
 - `void function(int numBytes)`
 - Número de datos disponibles `(Wire.available())`
 - Recepción de datos `(Wire.read())`

Maestro Lector → Esclavo Escritor

- Función de inicialización de la comunicación
`(Wire.begin(address))`
- Dispositivo Maestro:
 - Petición de datos `(Wire.requestFrom(address))`
 - Número de datos disponibles `(Wire.available())`
 - Recepción de datos `(Wire.read())`
- Dispositivo Esclavo:
 - Instalación de manejador `(Wire.onRequest(function))`
 - `void function(void)`
 - Envío de datos `(Wire.write(value/string/data,length))`

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/>