

Sistema low cost de comunicación Bluetooth Low Energy



Autor: Julia León Jiménez
Tutor: Aleix López Martín



Índice

1. Introducción

- a. Motivación
- b. Objetivos

2. Estado del Arte

- a. Bluetooth Low Energy
- b. Sistemas Desarrollados

3. Estructura Hardware

- a. Elección componentes
- b. Hardware Adicional
- c. Esquema General
Sistema de Comunicación

4. Estructura Software

- a. Sketch Arduino
- b. Firmware nRF52 DK

5. Pruebas Prototipo

- a. Conectividad
- b. Sistema Completo
- c. Aumento de Temperatura

6. Conclusiones

- a. Presupuesto
- b. Mejoras Futuras

Introducción – Motivación y Objetivos



Motivación

La tecnología **Bluetooth** integrada en los dispositivos portátiles comerciales permite que se comuniquen con otros dispositivos sin hacer uso de internet o red y, por tanto, sin coste

La tecnología **IoT**, que se encuentra en pleno auge, hace más interesante el diseño del sistema de comunicación low cost

Arduino y placas compatibles con Arduino

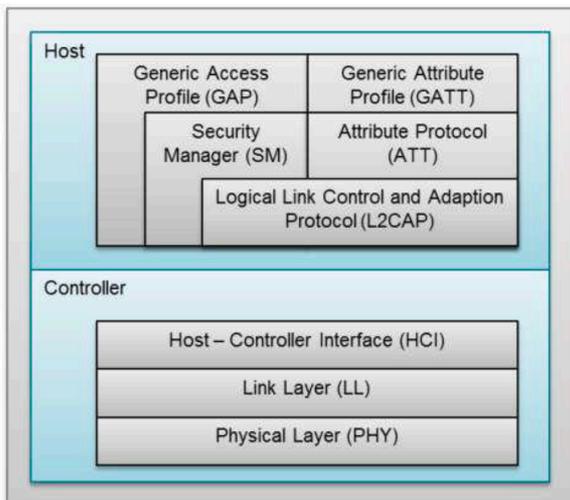
Objetivos

- Adquirir conocimiento de Bluetooth Low Energy
- Conocer los System on Chip BLE más comunes
- Desarrollo del software del sistema de comunicación
- Desarrollo de un prototipo y realizar pruebas
- Conclusiones: Coste y mejoras futuras

Estado del Arte – Bluetooth Low Energy



- Es un subconjunto del estándar Bluetooth que se caracteriza por su **bajo consumo**
- **Aparece a partir de la versión 4.0** del estándar Bluetooth
- La versión **Bluetooth 5.2 LE Audio** es la última versión del estándar Bluetooth

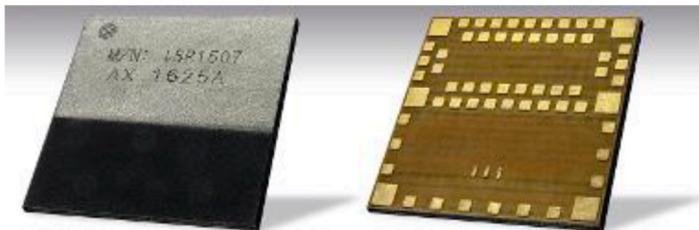


Características	Bluetooth Classic	BLE
Finalidad	Conexión entre dispositivos móviles, accesorios que requieren mayor energía	IoT, M2M, conexión con sensores y accesorios de baja energía
Tasa de transferencia	24Mb/s	1Mb/s
Consumo medio de energía	1W	0.01W-0.5W (depende del caso de uso)
Alcance máximo	100m	<100m
Tiempo de conexión	100ms	6ms
Modo de suspensión	-	Sleep Mode: Sólo se despierta cuando se recibe una solicitud de conexión



Estado del Arte – Sistemas Desarrollados

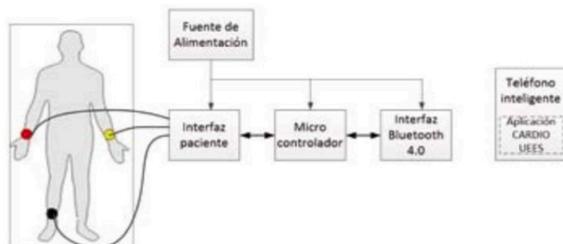
Módulo Bluetooth de Baja Energía ISP1507



Wearables



Sistema de Monitorización Cardíaca



Control de Iluminación de interiores



Estructura Hardware – Elección de Componentes

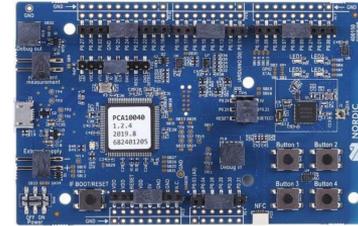


Arduino MEGA



Microchip	ATmega2560
Alimentación (V)	5V
Pines digitales	54 (15 PWM)
Pines analógicos	16 ($I_{m\acute{a}x} = 40mA$)
Memoria SRAM	8kB
Memoria Flash	256kB
Memoria EEPROM	4kB

nRF52 Development Kit – Nordic Semiconductor



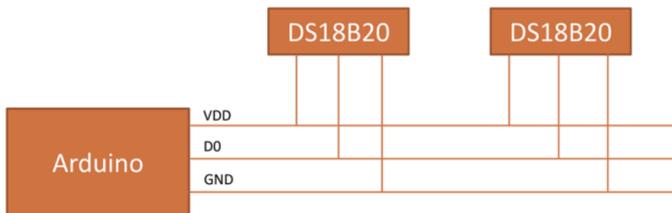
Pines I/O	32
Interfaces	SPI, I2C, UART, PWM, ADC
Memoria RAM	64kB
Memoria FLASH	512kB
Protocolos soportados	BLE, NFC, ANT

Estructura Hardware – Hardware adicional necesario

Sensor de temperatura DS18B20



Alimentación	3V – 5.5V
Precisión en la medida	$\pm 0.5^{\circ}\text{C}$ en el rango $[-10, 85]^{\circ}\text{C}$
Rango de medida	$[-55, 125]^{\circ}\text{C}$
Protocolo	1-wire
Precio	1€

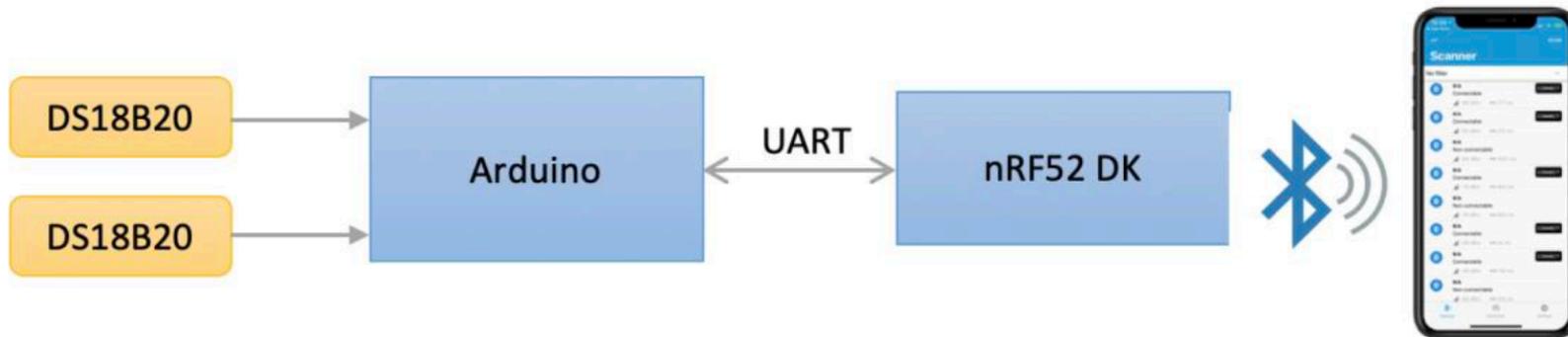


Smartphone – Aplicación nRF Connect



- Aplicación disponible para iOS y Android
- Propia del fabricante Nordic Semiconductor
- Permite **buscar, configurar y establecer comunicación** con dispositivos Bluetooth

Estructura Hardware – Esquema General del Sistema de comunicaci3n



Estructura Software – Sketch de Arduino

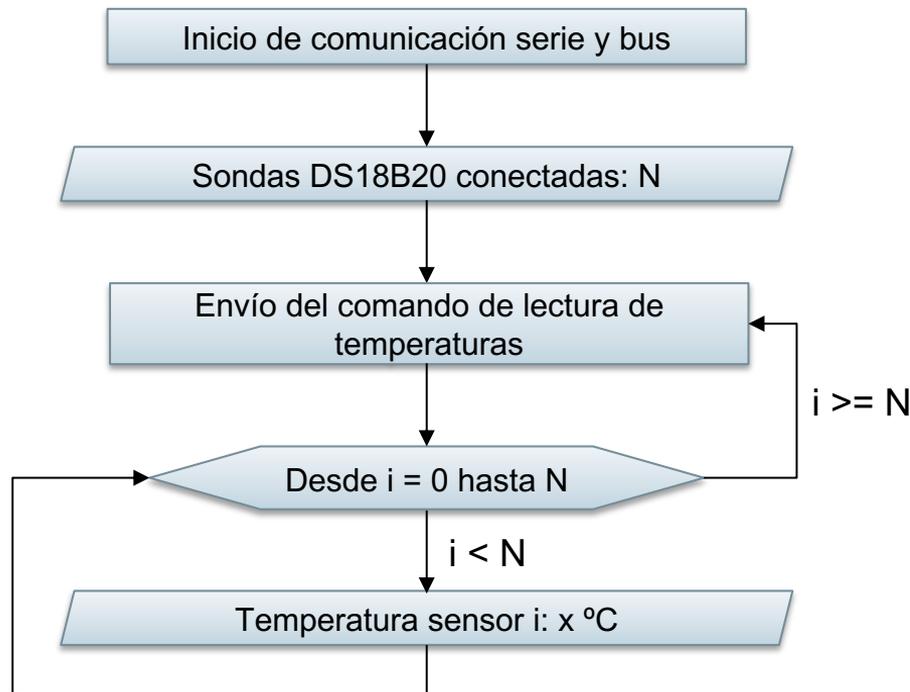


Librerías utilizadas

- SoftwareSerial.h
- DallasTemperature.h
- OneWire.h

Funciones utilizadas

- SoftwareSerial:
 - begin()
 - println()
- DallasTemperature:
 - begin()
 - getDeviceCount()
 - requestTemperatures()
 - getTempCByIndex()





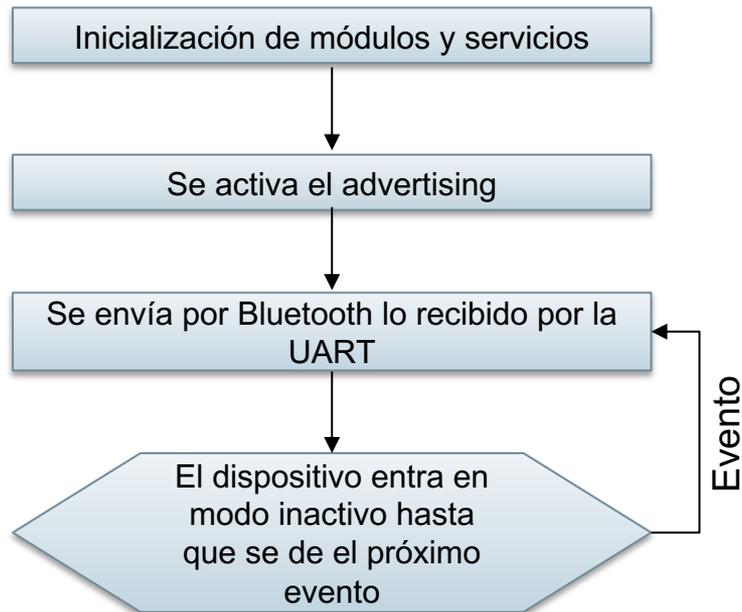
Estructura Software – Firmware nRF52 DK

Se parte de uno de los firmware dados en la página del fabricante

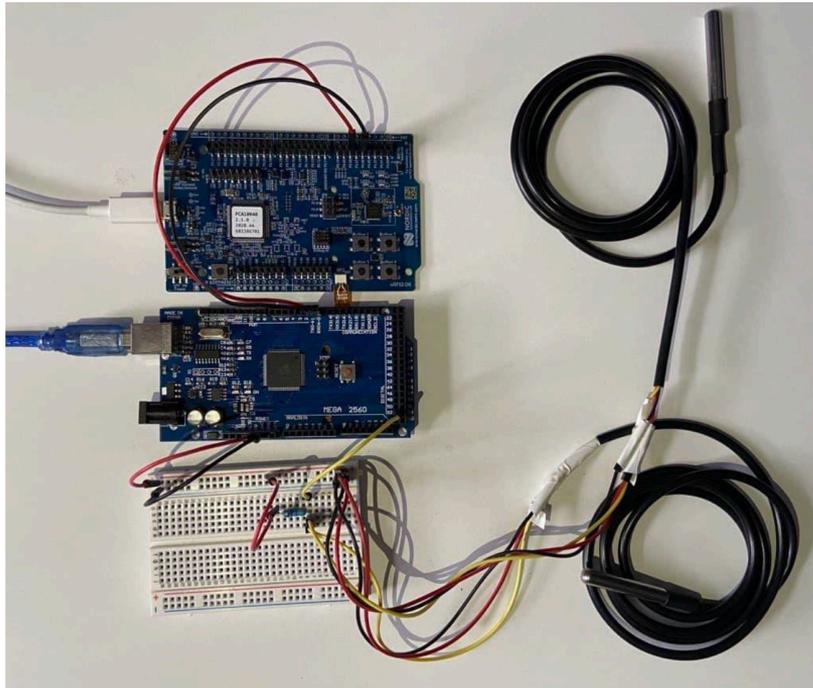
Se inicializan el log de nRF, la UART, los parámetros de conexión, los timers y los servicios a utilizar

Se inicializan los módulos relacionados con Bluetooth: GAP, GATT, advertising y stack BLE

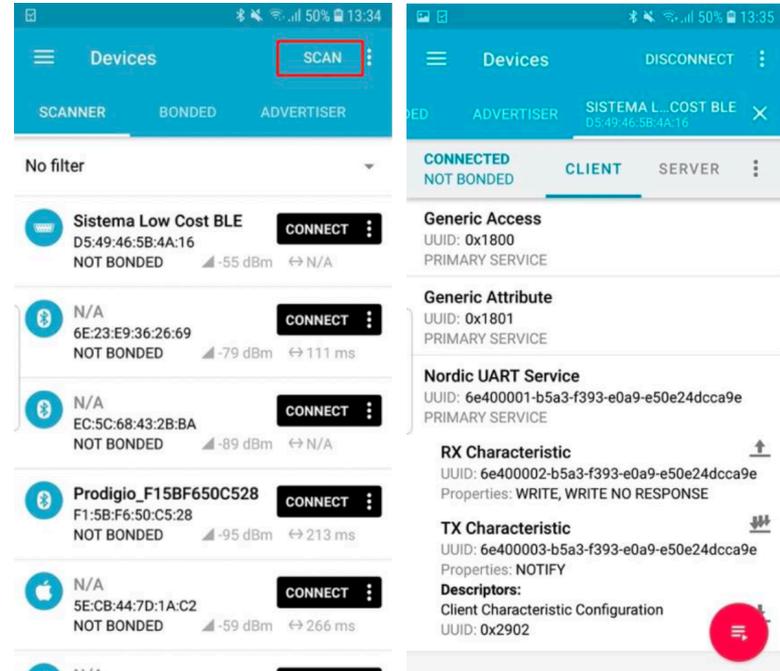
Se activa el modo de publicación y **se envía por Bluetooth la información recibida por la UART**



Pruebas Prototipo – Prueba de Conectividad



Escaneo y conexión al nRF52 DK desde nRF Connect



Pruebas Prototipo – Prueba de Conectividad



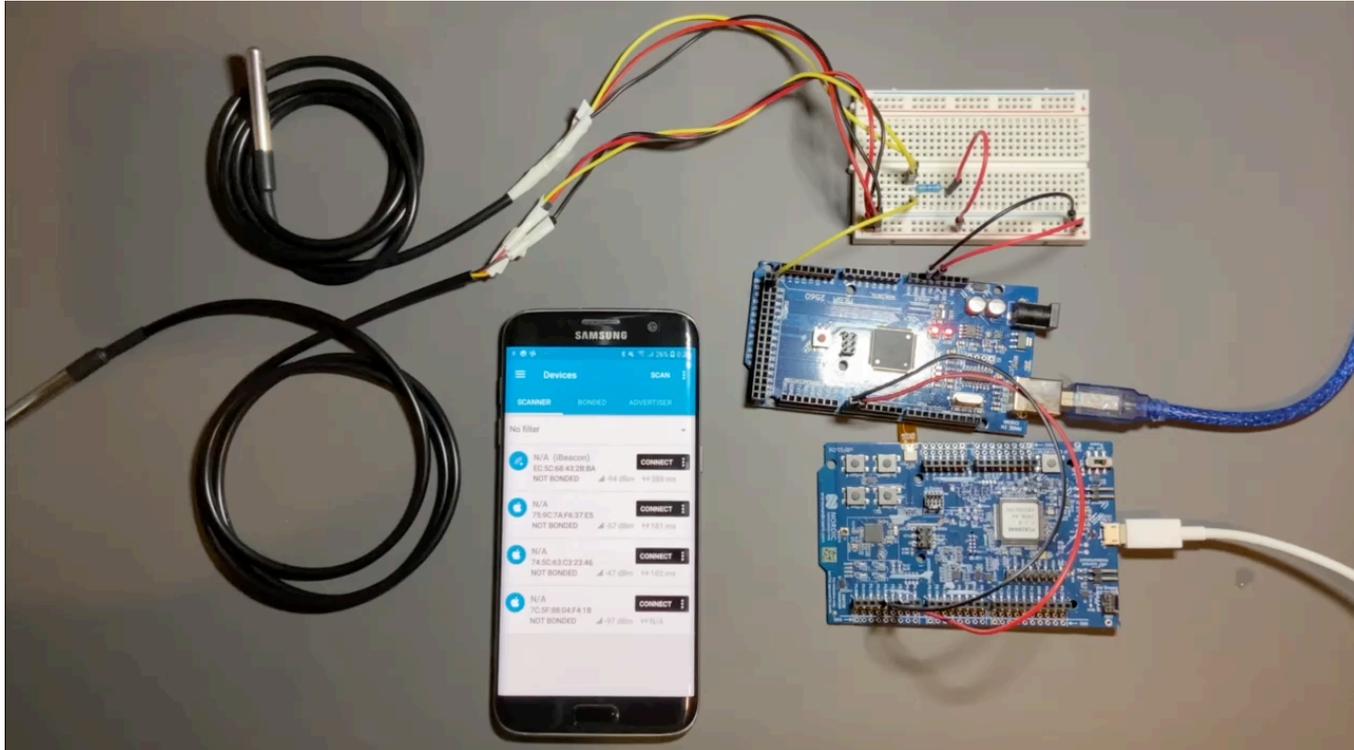
Activación de notificaciones

Apertura del log de la aplicación

The left screenshot shows the 'Generic Access' service configuration. The 'Enable CCCDs' option is highlighted with a red box. The right screenshot shows the 'Nordic UART Service' configuration. The 'Notifications enabled' option is highlighted with a red box.

The left screenshot shows the 'Bond' dialog with the 'Show log' option highlighted in a red box. The right screenshot shows the application log with several entries, including 'Notifications enabled sent' and temperature sensor notifications.

Pruebas Prototipo – Prueba del Sistema Completo



Conclusiones



Presupuesto

Componente	Coste
SEgger Embedded Studio	Gratuito
nRF5 SDK	Gratuito
Arduino MEGA	7.43€
nRF52 DK	33.66€
Sondas DS18B20	1€/ud
Componentes Arduino	1.94€
Total	45.03€

Mejoras futuras

- Añadir sensores/actuadores para supervisar y controlar hogares o parámetros de un ecosistema
- Optimización de la estructura software
- Desarrollo de una aplicación propia para la supervisión y control de los sensores y actuadores
- Desarrollo de un sistema de alarmas

Muchas gracias por su atención

 [UOC.universitat](https://www.facebook.com/UOCuniversitat)

 [@UOCuniversidad](https://twitter.com/UOCuniversidad)

 [UOCuniversitat](https://www.instagram.com/UOCuniversitat)

UOC