



# **MATRIX** | EBLOCKS 2

## **Bluetooth Communications**



CP1795

**MATRIX**  
[www.matrixtsl.com](http://www.matrixtsl.com)

Copyright © 2018 Matrix Technology Solutions Limited

CP1795

# Comunicación Bluetooth

Notas del  
curso

# Contenido

1	Introducción	6
1.1	Estructura de estas notas	6
1.2	Resultados del aprendizaje	6
1.3	Utilización de este manual	7
1.3.1	Introducción a las secciones de aplicación práctica	7
2	¿Qué entendemos por Bluetooth?	8
2.1	El módulo Bluetooth RN46788	
2.2	Modo de mando	8
2.3	Hardware y software utilizados en el curso	8
2.3.1	E-bloques	2 partes
2.3.2	Código de flujo	9
2.4	Dispositivos útiles adicionales	9
2.4.1	Dongle Bluetooth/USB integrado	9
3	Guesta en marcha	10
3.1	Configuración del hardware	10
3.1.1	Configuración de E-blocks	210
3.2	Introducción a Flowcode	11
3.2.1	Inicio en Flowcode	11
3.3	El componente Bluetooth	11
3.3.1	Propiedades y conexión de pines	12
3.3.2	Macros del componente Bluetooth	12
3.4	Pruebas del hardware	12
4	Teoría y antecedentes de Bluetooth	13
4.1	Introducción a Bluetooth	13
4.1.1	Breve historia	13
4.1.2	Conceptos de Bluetooth	13
4.1.3	Ventajas de Bluetooth	14
4.1.4	Desventajas de Bluetooth	14
4.2	Protocolos y modelo OSI	14
4.2.1	Aplicación	16
4.2.2	Capas Bluetooth	16
4.2.3	Detalles del hardware	17
4.2.4	Perfiles	19
5	Descubrimiento	20
5.1	Teoría: Encontrar otros dispositivos Bluetooth	20
5.1.1	Los comandos de investigación	20
5.1.2	Parámetros adicionales de la consulta	20
5.1.3	Ejemplo de descubrimiento	21
5.2	Ejercicio 1: Descubrir dispositivos Bluetooth	22
5.2.1	Introducción	22
5.2.2	Objetivos	22
5.2.3	Requisitos previos	22
5.2.4	Requisitos de hardware y software	22
5.2.5	Información sobre el ejercicio	22
5.2.6	Resultado del aprendizaje	23
5.2.7	Tareas adicionales	23
5.3	Aplicación práctica: Detección de dispositivos Bluetooth	23
5.3.1	Descubrir y ser descubierto	23
5.3.2	Planificación del programa	24
5.3.3	Macros necesarias	24
5.3.4	Inicialización	25

5.3.5 Envío de un comando	25	6.3.3 Creación del programa	32
5.3.6 Comprobación de respuestas	26	6.3.4 Ejecución y demostración del programa	33
5.3.7 Ejecución del programa completo	27	7 Conexión de dispositivos Bluetooth	34
6 Detectabilidad	28	7.1 Teoría: Conexión - Direcciones	34
6.1 Teoría: Descubribilidad	28	7.2 Ejercicio 3: Conexión a un dispositivo	35
6.1.1 Configuración para la puesta en marcha	28	7.2.1 Introducción	35
6.1.2 Uso de los comandos Set	28	7.2.2 Objetivos	35
6.1.3 Uso de los comandos Get	28	7.2.3 Requisitos previos	35
6.1.4 Uso de comandos de acción	28	7.2.4 Requisitos de hardware y software	35
6.1.5 Hacer que un dispositivo sea detectable	29	7.2.5 Información sobre el ejercicio	35
6.1.6 Entrar en modo comando	29	7.2.6 Resultado del aprendizaje	35
6.1.7 Esperar a que se ejecute la orden	29	7.2.7 Otros trabajos	35
6.1.8 Velocidad en baudios y ajustes	29	7.3 Aplicación práctica: Conexión	36
6.1.9 Modo autenticación	29	7.3.1 Reiniciar los sistemas	36
6.1.10 Modo esclavo	29	8 Claves de acceso y conexión	37
6.1.11 Activar la configuración actualizada	29	8.1 Teoría: Claves de acceso y conexión	37
6.1.12 Espere a que se reinicie y salga del modo de comandos	29	8.1.1 Envío del comando Passkey	37
6.1.13 Resumen de los comandos básicos de configuración	29	8.1.2 Iniciar el emparejamiento y la conexión	37
6.2 Ejercicio 2: Descubribilidad	30	8.2 Ejercicio 4: Claves de acceso y conexión	38
6.2.1 Introducción	30	8.2.1 Introducción	38
6.2.2 Objetivos	30	8.2.2 Objetivos	38
6.2.3 Requisitos	30	8.2.3 Requisitos previos	38
6.2.4 Requisitos de hardware/software	30	8.2.4 Requisitos de hardware y software	38
6.2.5 Información sobre el ejercicio	30	8.2.5 Información sobre el ejercicio	38
6.2.6 Resultado del aprendizaje	30	8.2.6 Resultado del aprendizaje	38
6.2.7 Tareas adicionales	30	8.2.7 Otros trabajos	38
6.3 Implementación práctica: Descubribilidad	31	8.3 Aplicación práctica: Passkeys y Connecting	39
6.3.1 Continuación del desarrollo	31	8.3.1 El programa básico	39
6.3.2 Configuración del dispositivo Bluetooth	31	8.3.2 Funciones avanzadas: Elegir el dispositivo al que conectarse	40
		8.3.3 Reiniciar los sistemas	40
		9 Comprobación de las respuestas	41

9.1 Teoría: Comprobación de respuestas41 9.1.1Respuestas solicitadas y no solicitadas41 9.1.2Macros de gestión de respuestas41 9.2Ejercicio 5: Comprobación de respuestas43

9.2.1Introducción43

9.2.2Objectives43

9.2.3Pre-requisites43

9.2.4Requisitos de hardware y software43 9.2.5Información sobre el ejercicio43 9.2.6Resultado del aprendizaje43

9.3 Aplicación práctica: Comprobación de las respuestas44

9.3.1 Uso de la macro WaitForStringValue44

9.3.2 Metodología de comprobación de errores44

10Introducción a Bluetooth Low Energy (BLE)45

10.1 Modos de funcionamiento de Bluetooth Low Energy (BLE)45

10.2 Ejercicio 6: BLE GAP transparente UART serial data ser- vice46

10.2.1 Introducción46

10.2.2Objetivos46

10.2.3Requisitos previos46

10.2.4Requisitos de hardware/software46

10.2.5Información sobre el ejercicio46

10.2.6Resultado del aprendizaje46

10.3 Implementación práctica: Servicio de datos en serie BLE47 10.3.1Modo BLE47

10.3.2Emparejamiento simple seguro47 10.3.3Activación de los ajustes47

11Confianza y seguridad48 11.1Teoría:

Confianza y seguridad48

11.1.1Seguridad en general48

11.1.2Modos de seguridad49

11.1.3Autenticación49

11.1.4Confianza49

11.2 Ejercicio 7: Confianza y seguridad50

11.2.1 Introducción50

11.2.2 Objetivos50

11.2.3 Requisitos previos50

11.2.4 Requisitos de hardware y software50

11.2.5 Información sobre el ejercicio50

11.2.6 Resultado del aprendizaje50

11.3 Aplicación práctica: Confianza y seguridad51

11.3.1 Objetivos generales51

11.3.2 Características adicionales51

12 Principios de diseño del proyecto52

12.1 Proyecto - Mando a distancia sencillo52

12.1.1 Ampliación del proyecto52

12.2 Proyecto - Datalogger médico52

12.2.1 Ampliación del proyecto52

# PARTE 1: Introducción general y visión de conjunto

## Introducción

### Estructura de estas notas

Estas notas se exponen a continuación:

Parte 1: Introducción general a Bluetooth y al kit de comunicación Bluetooth

- Primeros pasos: introducción al hardware y al software

Historia y visión general de Bluetooth

Parte 2: El curso

Una serie de ejercicios progresivos para llevar a los estudiantes a través de los conceptos y la práctica necesaria en el establecimiento de comunicaciones Bluetooth.

Los capítulos se dividen en 3 secciones para cada capítulo.

- Sección de teoría
- Descripción del

ejercicio Notas  
prácticas

Parte 3: Referencias y Apéndice

Capítulos complementarios con información de referencia. Incluye:

- Sección de referencia

Una sección de referencia "Cómo programar

## Resultados del aprendizaje

Estos apuntes del profesor están diseñados para introducir los conceptos y estrategias necesarios para las comunicaciones Bluetooth prácticas. Al completar los ejercicios de este curso, los estudiantes aprenderán lo siguiente:

- Cómo un dispositivo Bluetooth descubre a otro dispositivo Bluetooth y las opciones relativas a la capacidad de detección
- Cómo se emparejan los dispositivos Bluetooth y se establece un canal de comunicación
- Cómo se transfieren datos de diversos tipos entre dispositivos Bluetooth

Cómo gestiona Bluetooth la confianza y la seguridad.

Estas notas no abordarán las características de radiofrecuencia ni las características y protocolos de transmisión de bajo nivel de Bluetooth. De todo ello se encarga un módulo Bluetooth comercial que protege a los usuarios de estos problemas.

Estas notas están estructuradas en una serie de secciones que primero te llevan a través de la instalación, configuración y pruebas de hardware y software en el fondo de Bluetooth y luego en una serie de ejercicios y ejemplos que llevan al estudiante a través del funcionamiento de Bluetooth.

Los ejercicios deben realizarse utilizando Flowcode V10 o posterior, un lenguaje de programación gráfica. El componente Bluetooth de Flowcode está diseñado para que los estudiantes aprendan sobre Bluetooth sin tener que enfrascarse en los problemas de la programación en C o en un lenguaje de bajo nivel.

## Utilización de este manual

La parte principal de este manual se estructura en torno a un planteamiento en tres partes:

- **Teoría.** La primera parte de un capítulo introduce el tema en cuestión y analiza la teoría que lo sustenta, explica los comandos utilizados y las secuencias y estrategias generales necesarias.
- **Ejercicio.** El segundo elemento es un ejercicio. El ejercicio se presenta aquí para que se comprendan y se tengan en cuenta las metas y los objetivos durante la lectura de la sección siguiente.

**Aplicación práctica.** La tercera sección trata de la realización práctica del ejercicio. Se incluyen aquí las macros Flowcode utilizadas, las estrategias Flowcode y cualquier otra información o consejo necesario para alcanzar los objetivos del ejercicio.

Se pretende que los alumnos reciban primero una explicación teórica en forma de conferencia o folleto. A continuación, los alumnos pueden realizar el ejercicio. Los supervisores tienen la opción de entregar o no las notas de aplicación práctica.

Sugerimos que, para cada ejercicio, los estudiantes reciban la(s) hoja(s) de ejercicios y las notas de implementación práctica. Los estudiantes deben construir los programas en Flowcode. Las notas de implementación práctica iniciales son bastante detalladas y proporcionan mucha información sobre cómo debe construirse el programa. Los ejercicios posteriores no incluyen notas de implementación práctica tan detalladas: el alumno debe utilizar sus conocimientos para completar las tareas detalladas en el ejercicio.

La hoja de datos del módulo Bluetooth es una parte clave de la solución Bluetooth: se espera que los estudiantes busquen el significado de los comandos y funciones del módulo Bluetooth. En consecuencia, los estudiantes deben recibir una copia de la guía del usuario del módulo Bluetooth que se proporciona en el archivo zip de recursos CP1795.

Para la mayoría de los ejercicios se proporciona un conjunto completo de programas de solución. Se trata de dos programas, uno para cada nodo Bluetooth. Tenga en cuenta que estos ejemplos requerirán el cambio de cualquier ID de dispositivo utilizado dentro del programa.

### Introducción a las secciones de aplicación práctica

Esta primera sección práctica se presenta en forma de ejemplo con toda la información práctica y un programa de ejemplo. Las secciones prácticas posteriores contendrán la información adicional necesaria para lograr los objetivos del ejercicio, y contendrán fragmentos de código para nuevos procedimientos y macros, pero no incluirán necesariamente programas de ejemplo completos. En esta fase, se supone que los estudiantes son lo suficientemente competentes como para crear los programas a partir de la información proporcionada en la sección práctica.

Se pretende que, siempre que sea posible, los programas creados por el alumno para el ejercicio anterior se reutilicen para el ejercicio actual. Esto ilustra tanto la evolución del programa a medida que se añaden nuevos pasos, como para mostrar la creciente complejidad y el flujo del programa de un sistema de trabajo completo. Este enfoque también proporciona a los estudiantes un núcleo de código con el que ya están familiarizados. Sin embargo, los programas de los estudiantes pueden necesitar ser evaluados en varios puntos para asegurarse de que el código es adecuado para el proyecto actual, y las modificaciones o comentarios realizados en su caso para dirigir los programas de los estudiantes a través del curso en su conjunto. Dichas modificaciones pueden consistir en adaptar partes de los programas a macros para despejar partes del programa, o en reescribir el código de forma supervisada y evaluada para implementar elementos de forma más organizada y eficiente.

## ¿Qué entendemos por Bluetooth?

Se puede acceder a Bluetooth a varios niveles. Bluetooth es tanto una tecnología de comunicación como una estrategia de comunicación. Como tecnología de comunicaciones, Bluetooth transforma los datos de una aplicación en señales de radio y viceversa. Como estrategia de comunicación, Bluetooth consiste en descubrir y enlazar con otros dispositivos Bluetooth y acceder a las funciones de esos dispositivos.

Este curso se centrará en las estrategias de comunicación de Bluetooth. Aquí es donde el usuario final experimentará las tecnologías Bluetooth, como el envío de datos entre dispositivos.

La tecnología de comunicaciones Bluetooth suele ser gestionada automáticamente por los dispositivos Bluetooth. A menos que se diseñen activamente chips Bluetooth, las capas de comunicaciones inferiores permanecerán ocultas para la mayoría de los usuarios finales. La historia y los antecedentes se han proporcionado para mostrar cómo Bluetooth maneja las comunicaciones.

### El módulo Bluetooth RN4678

El dispositivo utilizado en la placa Matrix Bluetooth BL0170 es un dispositivo Microchip RN4678. El dispositivo es un módulo Bluetooth autónomo con el que se puede comunicar mediante un conjunto de comandos a través de una UART serie. Este dispositivo permite a los usuarios conectarse y utilizar otros dispositivos Bluetooth a nivel de aplicación. El uso de este dispositivo permite que el curso se concentre en las estrategias y secuencias de comunicación entre dispositivos Bluetooth sin necesidad de profundizar en los niveles tecnológicos inferiores.

### Modo de mando

El RN4678 funciona en dos modos: Modo de datos (por defecto) y modo de comandos.

Cuando el RN4678 está conectado a otro dispositivo y en modo Datos, el RN4678 actúa como una tubería de datos: todo lo que se recibe de la UART se pasa al dispositivo homólogo conectado a través de SPP si está conectado a un dispositivo Bluetooth Classic, o a través de un servicio GATT privado si está conectado a un dispositivo BLE. Cuando se reciben datos del dispositivo par desde SPP para Bluetooth Classic o UART Transparent para BLE, dichos datos salen directamente a UART.

El RN4678 se configura o controla, o ambas cosas, poniéndolo en modo comando y ejecutando comandos ASCII a través de UART.

..

### [Hardware y software utilizados en el curso](#)

El curso de Bluetooth utiliza el siguiente hardware y software:

#### Piezas E-blocks2

El kit de formación E-Blocks2 contendrá dos kits completos de los siguientes componentes:

- BL0011 Programador PIC o BL0055 Programador Arduino
- BL0170 Placa Bluetooth
- BL0169 Pantalla LCD

Y uno de cada

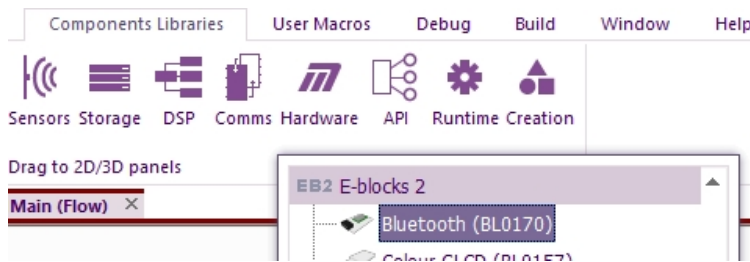
- BL0145 Placa de interruptores
- BL0167 Placa LED



# Flowcode

Los programas de ejemplo utilizados en este curso requieren Flowcode V10 o posterior.

Para Flowcode V10, el componente **Bluetooth BL0170** se encuentra en **E-blocks2** en la sección **Hardware** de la barra de herramientas **Bibliotecas de componentes**:



Todos los archivos de programas de ejemplo y ejercicio suministrados en el archivo zip de Recursos CP1795 han sido creados para Flowcode V10 o posterior.

Este curso asume un grado de familiaridad con Flowcode. Si es necesario, se debe dedicar tiempo utilizando los tutoriales y el curso Flowcode para familiarizar al estudiante con el uso de Flowcode.

## Dispositivos adicionales útiles

Los siguientes dispositivos no se suministran con la solución Bluetooth, sin embargo, si están disponibles, pueden utilizarse junto con la solución Bluetooth

### Dongle Bluetooth/USB integrado

Para las comunicaciones hacia y desde el PC se necesita un módulo Bluetooth. Algunos portátiles y PC recientes ya están preparados para Bluetooth. Sin embargo, la mayoría de los PC y portátiles actuales necesitan un módulo Bluetooth que se conecta a un puerto USB libre y permite que el ordenador funcione como un dispositivo Bluetooth.

Tenga en cuenta que no es necesario tener un módulo Bluetooth en su PC para completar todos los ejercicios.

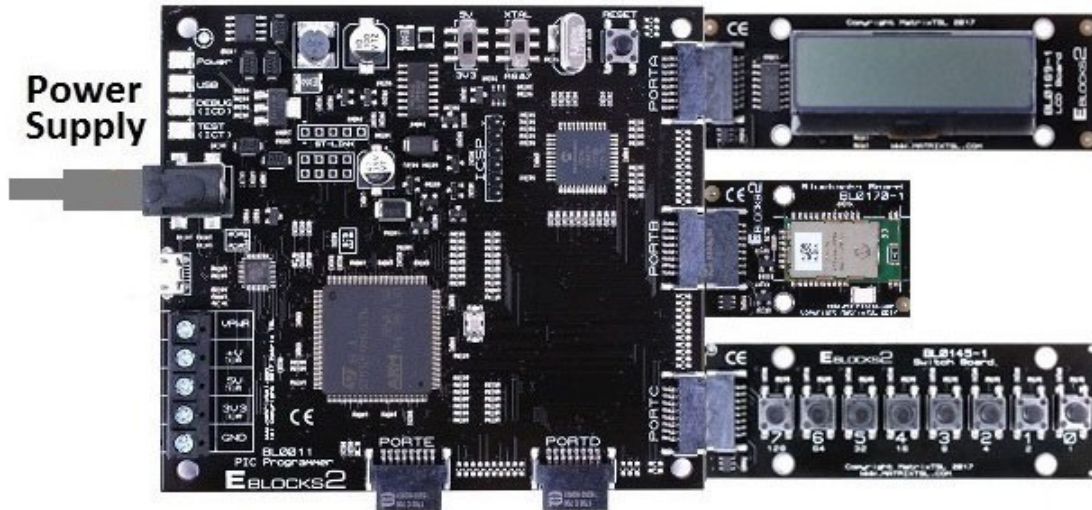
# Para empezar

## Configuración del hardware

### Configuración de E-blocks2

Los dos conjuntos de placas E-blocks2 deben conectarse como se muestra a continuación.

Algunos ejercicios utilizan una placa de conmutación sólo en el Nodo B, el nodo activo, como se muestra.



PUERTO	BL0011 Programador PIC
PUERTO A	BL0169 Placa LCD
PUERTO B	BL0170 Tarjeta Bluetooth
PUERTO C	Algunos proyectos utilizan la placa de interruptores BL0145 o la placa de LEDs BL0167



PUERTO	BL0055 Programador Arduino
PUERTO A0-5	BL0169 Placa LCD
PUERTO D0-7	BL0170 Tarjeta Bluetooth
PUERTO D8-13	Algunos proyectos utilizan la placa de interruptores BL0145 o la placa de LEDs BL0167

Flowcode es un sistema de diagramas de flujo para microcontroladores. Por lo general, los microcontroladores se programan en C (un lenguaje difícil de aprender) o en ensamblador (un lenguaje aún más difícil de programar). La enseñanza de los microcontroladores suele requerir una gran inversión de tiempo para que los alumnos adquieran los conocimientos básicos de los lenguajes necesarios antes de que puedan enfrentarse a sistemas como Bluetooth.

Flowcode es un método mucho más sencillo e intuitivo de crear programas. Basado en símbolos de diagramas de flujo estándar y utilizando iconos de arrastrar y soltar, se puede construir un programa de trabajo en cuestión de minutos. Los iconos pueden configurarse mediante cuadros de diálogo que eliminan la posibilidad de que se produzcan errores de sintaxis o se seleccionen opciones no válidas. Los diagramas de flujo pueden seguirse y rastrear visualmente, lo que permite a los usuarios ver la imagen más amplia, así como el elemento individual. La mayoría de los cursos de introducción a la programación utilizan o recomiendan la creación de diagramas de flujo como paso previo a la escritura del código final, debido a su capacidad para desglosar el flujo del programa de una manera clara y comprensible. Con Flowcode eso ES escribir su programa.

Se puede utilizar una amplia gama de componentes con programas que van desde LEDs e interruptores básicos hasta sistemas de comunicación completos como CAN, TCP/IP y, por supuesto, Bluetooth. Una vez más, los diálogos se utilizan para eliminar la posibilidad de errores de sintaxis o de llamadas a funciones mal realizadas. Los iconos de macro permiten al usuario acceder a las funciones de los componentes para realizar tareas tan variadas como encender un solo LED, enviar texto a una pantalla LCD o comprobar si hay mensajes entrantes en un sistema Bluetooth.

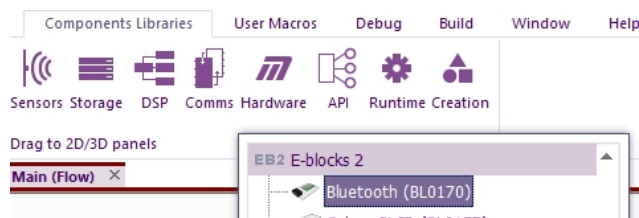
El sistema se ha diseñado para que sea fácil de entender y utilizar, pero con la profundidad y flexibilidad que exigen los mercados técnicos, educativos e industriales actuales.

## Introducción a Flowcode

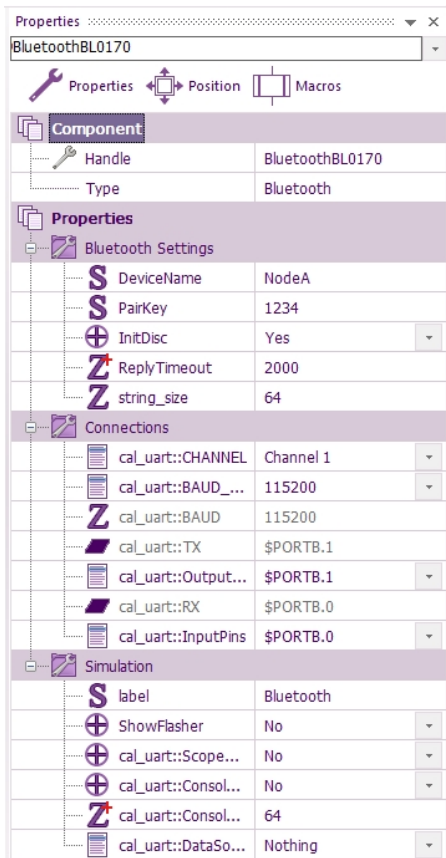
Este curso asume un grado de familiaridad con Flowcode. Los nuevos usuarios de Flowcode deben consultar los tutoriales de Flowcode y el curso introductorio antes de utilizar Flowcode con este curso, que se pueden encontrar en el sitio web [www.flowcode.co.uk](http://www.flowcode.co.uk). Consulte la página web de Matrix TSL [www.matrixtsl.com](http://www.matrixtsl.com) y el Centro de Aprendizaje de Matrix para obtener información y enlaces sobre los cursos.

## El componente Bluetooth

Para Flowcode V10, el componente **Bluetooth BL0170** se encuentra en **E-blocks2** en la sección **Hardware** de la barra de herramientas **Bibliotecas de componentes**:



Añadir este componente al Panel del sistema



El componente Bluetooth BL0170 se basa en el dispositivo RN4678 y dispone de una serie de propiedades y conexiones que el usuario puede configurar.

Entre ellos se incluyen el nombre del dispositivo, la clave de par y si el dispositivo es detectable o no. Estos ajustes se aplican cuando se inicializa el dispositivo.

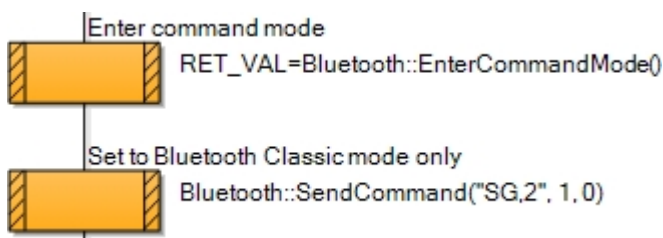
Aquí también se configuran los detalles de la conexión UART serie al dispositivo.

## Macros de componentes Bluetooth

El componente Bluetooth utiliza una serie de macros (o funciones) para ayudar a programar el dispositivo Bluetooth. Encontrará información detallada sobre las macros, sus parámetros y sus usos en el archivo de ayuda del componente Bluetooth.

La sección de ejercicios didácticos del curso introducirá y explicará el uso de las macros a medida que se necesiten durante los ejercicios.

Como en el siguiente ejemplo, que fuerza al dispositivo Bluetooth a entrar en modo comando y luego emite un comando.



## Probar el hardware

Los kits de formación Bluetooth pueden probarse, y sus direcciones de dispositivo Bluetooth pueden determinarse, utilizando el programa de prueba Flowcode incluido llamado BT\_TEST,fcfx

Esto se puede encontrar en el CP1795 Recursos zip.

Conecte las tarjetas como se indica en la sección 3.1 y conecte la alimentación. Conecte cada kit a su vez al PC host que ejecuta Flowcode a través del cable USB y descargue el programa.

Si la prueba se realiza correctamente, la pantalla LCD mostrará la dirección del dispositivo del módulo Bluetooth conectado. Puede anotarla y utilizarla como dirección de conexión en ejercicios posteriores.

## Introducción a Bluetooth

### Breve historia

"El rey Harald mandó hacer este monumento para Gorm, su padre, y Thyri, su madre. Ese mismo Harald que ganó toda Dinamarca y Noruega, y convirtió a los daneses en cristianos". Así dice la Piedra Rúnica de Jelling. Este Harald no es otro que el legendario rey vikingo Harald Bluetooth, que da nombre a la tecnología Bluetooth. Bluetooth empezó siendo sólo el nombre de un proyecto, algo para llamar a la tecnología en desarrollo. Pero el nombre se quedó.

Y es un nombre muy apropiado. Al igual que el Bluetooth de la época vikinga unió a naciones dispares, les dio un nuevo credo y las condujo a la grandeza, el Bluetooth actual une dispositivos de hardware dispares, les da un nuevo protocolo y amplía sus capacidades. Sin embargo, el monumento actual a Bluetooth no serían unas runas grabadas en piedra, sino la ráfaga de señales electrónicas cuando el mundo se comunica a través de Bluetooth.

N.B. El logotipo de Bluetooth contiene las runas H y B de Harald Bluetooth:



Bluetooth fue introducido en 1998 por el Bluetooth Special Interest Group, una federación de empresas dedicadas a las comunicaciones, la industria y las tecnologías empresariales. La variedad y el número de empresas implicadas han contribuido al éxito de Bluetooth y a su adopción generalizada por distintos sistemas.

La versión 1.1 de la especificación Bluetooth original (las versiones 1.0 y 1.0B eran demasiado limitadas tecnológicamente para un desarrollo importante) tuvo un éxito inmediato. La especificación se actualizó a la versión 1.2 en 2003, con características añadidas como mayor velocidad y salto de frecuencia adaptativo. En 2004 se publicó la versión 2.0 + velocidad de datos mejorada (EDR). V2.0+EDR es más rápida, consume menos energía y permite mejores comunicaciones entre dispositivos multiplataforma. Bluetooth sigue creciendo y adaptándose a las cambiantes necesidades de telecomunicaciones del mercado moderno.

## Conceptos Bluetooth

- Bluetooth está diseñado para permitir la transferencia inalámbrica segura de datos entre dispositivos de hardware.
- Mediante el uso de un protocolo, los datos pueden transmitirse entre los dispositivos de manera estándar. Esto simplifica la codificación y las pruebas, ya que no es necesario crear código propio.
- Bluetooth está totalmente especificado, lo que permite a los fabricantes crear hardware compatible con Bluetooth sin necesidad de diseñar e implantar protocolos propios de alto nivel. Esto también permite simplificar la red, ya que no hay problemas con protocolos o especificaciones propios que compitan entre sí.
- La comunicación es automática una vez que dos dispositivos Bluetooth están dentro del alcance. Esto significa que los usuarios no necesitan pasar por procedimientos de configuración para empezar a comunicarse.

Los dispositivos Bluetooth pueden hacerse seguros y no detectables, lo que ayuda a eliminar las amenazas a la seguridad de la comunicación automática. Tú decides si quieres estar en la red o no.

Bluetooth es, en esencia, una red de comunicaciones inalámbricas capaz de descubrir y comunicarse automáticamente con los nuevos dispositivos que entran en su radio de acción. Esto ofrece dos ventajas fundamentales al usuario.

## Ventajas de Bluetooth

No se necesitan cables.

Sin espaguetis de cables por todas partes. Sin líos antiestéticos. Nada con lo que tropezar o dañar. Y para usos como la tecnología de automoción, sin mazos de cables que instalar y mantener/ reparar/ sustituir.

No hay limitaciones físicas debidas a la colocación de cables o puntos de red, por lo que los dispositivos tienen mayor libertad de movimiento.

La comunicación se basa en la distancia, por lo que los dispositivos pueden ser móviles.

Los dispositivos pueden pasar de una red a otra sin necesidad de infraestructura.

Detección automática de otros dispositivos dentro del alcance.

No es necesario realizar cambios físicos, como nuevos puntos de red, para dar cabida a nuevos dispositivos. Por tanto, facilidad de uso y facilidad de actualización.

Las funciones de seguridad de Bluetooth te permiten configurar el dispositivo para que responda a otros dispositivos o los ignore, lo que te pone a ti al mando y no a la red.

Se pueden añadir nuevos dispositivos simplemente acercándose a su alcance, por lo que no es necesario editar ni cambiar la configuración ni nada para añadir nuevos dispositivos.

## Desventajas de Bluetooth

Complejidad y fiabilidad

Puede que los cables no sean portátiles, pero son sencillos y fiables. La complejidad, como en los sistemas Bluetooth, conlleva un mayor riesgo de fallos.

Cuanto más complejo es un sistema, como Bluetooth, más difícil puede ser encontrar y solucionar fallos.

Duración de la batería

Los dispositivos móviles están limitados por la duración de la batería y las posibilidades de recarga. Puede producirse una pérdida de energía durante las comunicaciones.

Los cables pueden estar normalizados, pero las tomas de corriente y los niveles de tensión no. Si viaja, puede necesitar un adaptador de enchufe.

Costes

Los dispositivos Bluetooth suelen ser más caros que las soluciones fijas convencionales, como un cable. Puede ser necesario un desembolso inicial para pasar de los equipos heredados a las soluciones Bluetooth.

Mayores costes de reparación.

Velocidad

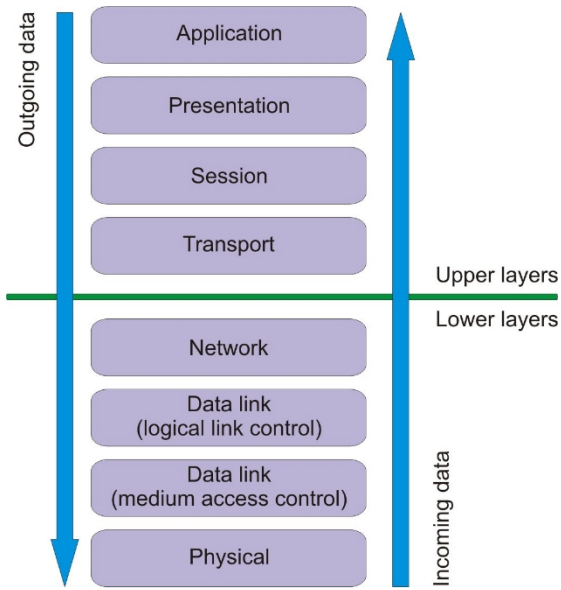
Bluetooth es más lento que las soluciones fijas equivalentes, como un cable directo.

La velocidad de las comunicaciones Bluetooth puede verse restringida por consideraciones relativas a la velocidad en baudios o por limitaciones de las velocidades de transferencia de datos del protocolo Bluetooth.

## Protocolos y modelo OSI

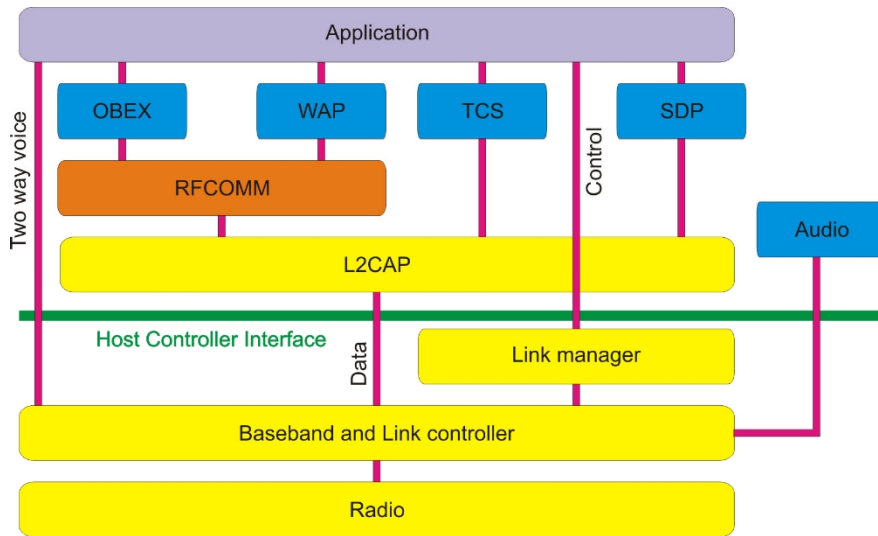
Bluetooth funciona en capas basadas en el modelo OSI. El modelo OSI es una secuencia de capas utilizadas para definir cómo los datos pasan de la aplicación al envío físico de señales. Este proceso se repite a la inversa tras la recepción de los datos para volver a pasar de las señales a los datos de la aplicación. Cada capa aborda áreas fundamentales de los procesos de comunicación. Al abstraer las diferentes partes del proceso de comunicaciones en capas basadas en el modelo OSI, los desarrolladores pueden implementar sus sistemas como una serie de tareas, cada una de las cuales cubre una o más capas OSI. Se pueden implementar estándares para tareas o capas específicas, creando una base sobre la que los desarrolladores pueden trabajar.

A continuación se muestra el diagrama básico del modelo OSI:



El modelo OSI se utiliza para ayudar a separar los distintos

elementos Las capas de Bluetooth se pueden ver a continuación:



i

El proceso básico es sencillo: los datos brutos de la parte superior deben convertirse en bits y bytes que puedan transmitirse por señal de radio y reconstruirse en datos brutos con los que pueda trabajar la aplicación del otro extremo.

La transmisión consiste en dividir los datos en fragmentos que puedan enviarse y envolverlos con información adicional que esa capa concreta pueda utilizar para comprobar errores o datos que falten. Una sola pieza de datos de una aplicación puede dividirse en varios trozos diferentes, cada uno de ellos rodeado por varias envolturas diferentes.

La recepción implica un proceso de "pasar el paquete" en el que cada capa desenvuelve la envoltura superior y pasa los paquetes a la capa especificada en esa envoltura. En cada paso pueden producirse otras funciones, como la comprobación de errores y el ensamblaje de grupos de paquetes en un paquete mayor que se pasará a la capa superior una vez completado. Finalmente, el paquete llegará a la aplicación y los datos podrán mostrarse al usuario (o pasar por el altavoz si es audio, etc.).

No es necesario utilizar todas las capas. Algunos paquetes pueden saltarse varias capas. El proceso depende del tipo de datos que se envíen.

## Aplicación

La aplicación es el elemento que crea o utiliza los datos, ya sean datos brutos, señales de audio o información textual. Es el nivel en el que los usuarios nos comunicamos con el sistema. Si utilizamos la oficina de correos como analogía de Bluetooth, la aplicación serían las palabras que escribimos o leemos.

Los datos que introducimos, ya sean hablados, mecanografiados, etc., se convierten en una forma adecuada para su transmisión, ya sean datos ASCII o datos en formato de audio. A continuación, los datos pueden enviarse a la cadena de transmisión.

Cuando se reciben los datos, están en un formato que la aplicación puede entender. Ya se ha hecho todo el cotejo de paquetes de datos y el reensamblaje. Son datos en bruto listos para ser utilizados.

## Capas Bluetooth

### **OBEX/WAP/TCS/SDP/AUDIO**

Se trata de protocolos de datos. En lugar de utilizar modelos de datos personalizados, es una buena práctica hacer uso de los protocolos y estándares existentes. Esto permite comunicarse con otras aplicaciones a través de los mismos protocolos estándar. Los datos se convierten al formato del protocolo listo para ser transmitido, o se convierten del protocolo a los datos brutos utilizados por la aplicación. El uso de estos protocolos comunes simplifica las tareas de programación y facilita la transferencia de código a otros dispositivos.

Algunos protocolos, como SDP y TCS, pueden utilizarse para ayudar a configurar las operaciones Bluetooth. SDP, por ejemplo, permite a las aplicaciones consultar al dispositivo Bluetooth qué servicios puede ofrecer.

Los protocolos más utilizados son:

OBEX - Intercambio de objetos

WAP - Protocolo de acceso inalámbrico

TCS - Especificación de control de

telefonía SDP - Protocolo de

descubrimiento de servicios

AUDIO - señal de audio directa como la utilizada para auriculares

### **RFCOMM**

RFCOMM, Radio Frequency **COMM**unication, es un protocolo especial. RFCOMM permite a los usuarios utilizar una conexión de puerto COM serie estándar en lugar de una conexión de radio Bluetooth. Se trata de puertos COM virtuales, no de puertos COM físicos. RFCOMM engaña al PC haciéndole creer que sus puertos COM virtuales existen de la misma forma que los puertos físicos reales. Pero cuando recibe datos los envía silenciosamente al enlace de radio Bluetooth. La aplicación final no se da cuenta; para ella es sólo un puerto serie al que envió datos. Cuando recibe datos, son sólo datos de un puerto serie COM, la parte Bluetooth está totalmente oculta.

La gran ventaja de esto es que la miríada de tecnologías de puerto COM serie existentes pueden utilizarse directamente sin necesidad de adaptadores especiales ni software totalmente nuevo. Esto puede tener un efecto considerable en el coste y el esfuerzo que supone actualizar o desarrollar productos que actualmente utilizan la tecnología de puerto COM serie. La mejor analogía de Correos para RFCOMM es el correo electrónico. Es una comunicación tipo carta, pero no es una carta. Los correos electrónicos tienen el mismo formato que una carta, los escribimos y los recibimos, pero el proceso de envío no lo gestiona Correos. Del mismo modo, el proceso de envío de datos en las comunicaciones serie RFCOMM no es realmente una comunicación serie aunque lo parezca al controlador host o al PC.

### **RFCOMM y el módulo RN4678**

El módulo RN4678 se sitúa entre la capa RFCOMM y la capa de aplicación. El módulo se encarga de la parte RFCOMM, pero requiere que se le envíen los comandos descritos en el conjunto de comandos, por lo que no se encuentra en el nivel de aplicación. Esto permite a los programadores



para concentrarse en la estrategia de comunicaciones y dejar los aspectos específicos de RFCOMM al módulo Bluetooth RN4678.

## **L2CAP**

El Controlador de Enlace Lógico y Protocolo de Adaptación controla y coordina el flujo de datos creando canales virtuales, sesiones y transferencias de archivos. El L2CAP es también la principal área de ensamblaje de paquetes que descompone los datos en paquetes individuales listos para ser enviados a la capa de banda base para su transmisión. En el viaje de vuelta, el L2CAP es la principal área de reensamblaje que coteja todos los paquetes individuales y vuelve a ensamblar los datos a partir de ellos. En nuestra oficina de correos de dientes azules, esta capa serían las páginas de la carta y el sobre en el que viene.

## **Interfaz del controlador host**

La interfaz de controlador de host enlaza el PC o el sistema con el hardware Bluetooth. Los elementos de la HCI incluyen los controladores de dispositivo necesarios para ejecutar el hardware Bluetooth y su interfaz. Donde el hardware Bluetooth se encuentra con el hardware de transferencia de datos es donde reside la HCI.

## **Gestor de enlaces**

El gestor de enlaces es un lenguaje de bajo nivel para configurar y controlar los enlaces.

Dada la sofisticación de las transmisiones Bluetooth, con saltos de frecuencia adaptables y una amplia gama de frecuencias disponibles, el gestor de enlaces es una herramienta muy útil. El gestor de enlaces ayuda a nivel de red y de enlace de datos. El gestor de enlaces es como el código postal que utiliza Correos para automatizar gran parte de su clasificación.

## **Controlador de banda base y enlace**

El controlador de banda base y enlace, correspondiente al nivel de enlace de datos del modelo OSI, es un controlador de paquetes. Reúne los datos que se le envían en paquetes de datos listos para su transmisión por la capa de radio. El contenido de los datos y quién los envía son irrelevantes. Lo único que importa es que la capa de banda base pueda ensamblarlos en paquetes de datos para su transmisión.

Se examinan las etiquetas de los paquetes de los datos que llegan desde la capa de radio y se envían a la capa adecuada. A la capa de banda base no le importa lo que contengan los datos, sino a quién entregárselos. En nuestra analogía con Correos, esta capa es la oficina de clasificación primaria y la dirección de la carta.

## **Radio**

La capa de radio es la capa física, el nivel más bajo disponible en Bluetooth. En este nivel, la única preocupación real es la señal de radio. Se establece una conexión de radio y se envían los datos. Los paquetes de datos ensamblados por la banda base se envían, y los paquetes de datos entrantes se reciben y pasan a la capa de banda base.

## **Detalles del hardware**

El hardware Bluetooth utiliza una serie de características de hardware y software comunes en sus comunicaciones que es útil conocer.

### **Piconets**

Un dispositivo Bluetooth puede formar parte de una red de hasta 8 dispositivos Bluetooth. La red está formada por un dispositivo denominado Maestro y hasta siete dispositivos Esclavos. Esta minired se denomina Piconet. Los esclavos sólo pueden enviar mensajes al dispositivo maestro, no a los demás esclavos. El Maestro se comunica con los Esclavos en un sistema 'Round robin' hablando con cada dispositivo Esclavo por turnos. Así, las señales pueden pasar a otro dispositivo esclavo a través del maestro, pero no directamente.

Los próximos dispositivos Bluetooth tendrán la capacidad de ampliar esto mediante la comunicación entre Piconets. Un dispositivo será maestro en un anillo y esclavo en otro, lo que permitirá la comunicación entre los dos anillos.

## **Buscapersonas e investigación**

Todos los dispositivos Bluetooth tienen la capacidad de convertirse en dispositivos Maestros o Esclavos. El dispositivo Maestro en una Piconet es el que inició el contacto, los Esclavos los que respondieron. El contacto es

se inicia enviando señales de consulta para determinar qué dispositivos están cerca. Al mismo tiempo, los dispositivos escuchan el contacto de otro dispositivo. Si se encuentra un dispositivo, se le puede llamar para establecer contacto y establecer una Piconet, o unirse a una si ya existe. Si existe un Maestro, como es el caso al unirse a una Piconet, el nuevo dispositivo será un Esclavo. Si no existe una Piconet, el dispositivo que inició el contacto se convierte en el dispositivo Maestro. Depende de quién haya recibido la señal de solicitud. Como los dispositivos pueden unirse o abandonar la red a voluntad, existen mecanismos para sustituir a un dispositivo maestro que abandona la Piconet.

### **Especificaciones de radio**

Bluetooth funciona en la banda de radio ISM de 2,45 GHz. La banda se divide en 79 canales diferentes entre los que Bluetooth salta 1600 veces por segundo (Especificaciones 1.1 y 1.2). Posteriormente, los dispositivos Bluetooth de especificación 2 pueden ser hasta 3 veces más rápidos. Bluetooth salta de frecuencia para evitar interferencias con otros sistemas que utilizan la misma frecuencia (por ejemplo, un vigilabebés) y para evitar que los dispositivos Bluetooth acaparen un canal concreto y lo bloqueen para otros dispositivos. Evidentemente, el sistema debe saber a qué canal cambiar. Esto se hace a nivel de radio, con el transmisor y el receptor siguiendo automáticamente el ritmo de los saltos de frecuencia de los demás dispositivos de la red. Las señales que se envían por radio contienen datos sobre el patrón de saltos de frecuencia que está utilizando el dispositivo y que los demás pueden utilizar para saber a cuál ir a continuación. Afortunadamente, todo esto se hace automáticamente en las capas inferiores, por lo que no es necesario configurar o programar la placa Bluetooth E-blocks2.

Debido a la baja potencia de la señal de radio (alrededor de 1 milivatio), el alcance está limitado a unos 9 metros, pero es más que suficiente para la mayoría de los dispositivos Bluetooth.

### **Transferencia de audio**

Debido a sus raíces en las telecomunicaciones, las señales de audio son una parte importante de muchos dispositivos Bluetooth.

Para simplificar la transferencia de audio, Bluetooth puede realizar un seguimiento rápido de las señales de audio digitalizadas a través del sistema hasta la salida de audio en el dispositivo receptor. Los datos de audio pueden enviarse a través de los métodos normales, y aplicaciones como un reproductor MP3 Bluetooth utilizarían generalmente la ruta normal de aplicación a aplicación, ya que tiene una mejor integridad de datos, y el almacenamiento en búfer de datos en el extremo receptor manejará la mayoría de los problemas de latencia. Para una comunicación directa, como un teléfono, es preferible la ruta directa, ya que la calidad de los datos -es decir, el ruido- es menos problemática que la latencia -es decir, la espera para oír lo que se dice-, ya que los datos tardan en recorrer el sistema. Las señales llegan directamente a las capas inferiores, se procesan y comprimen en el nivel de banda base y se envían al receptor, que envía los datos directamente desde su nivel de banda base al dispositivo de salida de audio.

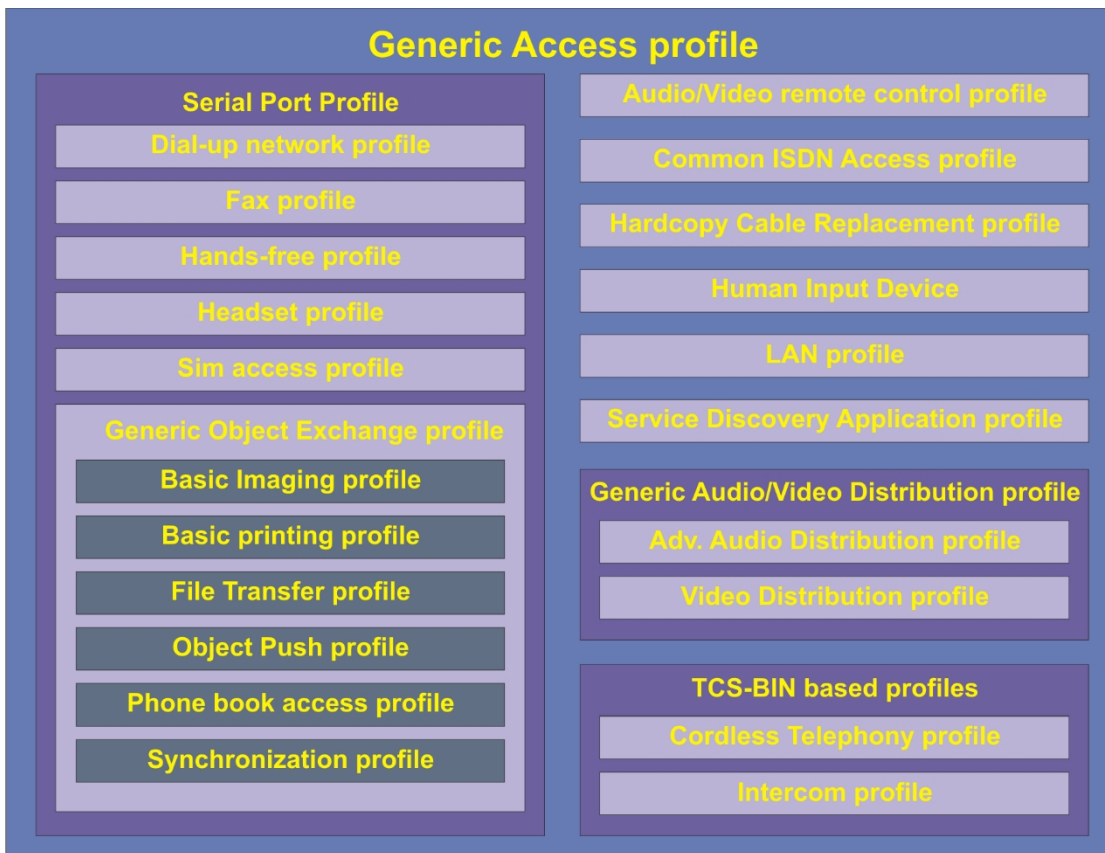
### **Autenticación y cifrado**

Bluetooth es un dispositivo de radio. Esto lo hace intrínsecamente inseguro, ya que cualquier dispositivo dentro del alcance de transmisión puede escuchar la señal. Por tanto, la seguridad es una de las principales preocupaciones de Bluetooth.

El principal nivel de seguridad es el emparejamiento. Los dispositivos deben emparejarse, para lo que es necesario conocer la dirección y la clave de acceso del dispositivo con el que se está emparejando. Los dispositivos pueden bloquear el acceso a la dirección, lo que imposibilita el emparejamiento, y la clave de acceso no es accesible mediante comunicaciones, por lo que debe conocerse para que el dispositivo pueda emparejarse. En la práctica, se trata de un número PIN suministrado con el producto. Por lo tanto, para emparejar un teléfono y unos auriculares, deberá introducir el número PIN de los auriculares en el teléfono cuando se le solicite. Sin PIN no hay comunicación.

Una vez emparejados, las comunicaciones pueden comenzar. Los dispositivos Bluetooth pueden configurarse para cifrar y autenticar las comunicaciones, asegurándose así de que los datos proceden de un dispositivo autorizado y están codificados, por lo que la escucha resulta infructuosa. Cualquier código puede ser descifrado, pero el cifrado Bluetooth y el corto alcance necesario para la escucha lo convierten en un sistema muy seguro.

# Perfiles



Los perfiles son implementaciones basadas en código para su uso con estructuras de hardware estándar, como auriculares o teléfonos móviles. La conformidad con un perfil permite al hardware hacer uso de características y comportamientos apropiados para ese tipo de dispositivo. Por ejemplo, el uso del perfil de auriculares permite que otras unidades se comuniquen con el dispositivo Bluetooth mediante llamadas estándar que forman parte de la implementación de auriculares. El perfil de Auriculares contendrá una serie de funciones que también están en una implementación de Teléfono, pero muchas de las funciones de Teléfono no estarán en el perfil de Auriculares. Esto se debe a que el perfil de auriculares es un subconjunto del perfil de teléfono.

La estratificación de perfiles permite a los perfiles de nivel superior utilizar las mismas funciones que los de nivel inferior. Por ejemplo, el perfil de fax puede utilizar las funciones de transferencia de archivos que forman parte del perfil de transferencia de archivos. También puede usar las características del perfil de Sincronización, ya que éste también es un subconjunto del perfil de Intercambio Genérico de Objetos, que es un subconjunto del perfil de Puerto Serie en el que reside el perfil de FAX.

## Parte 2: Ejercicios didácticos

### Descubrimiento

#### Teoría: Encontrar otros dispositivos Bluetooth

Para que un dispositivo Bluetooth pueda comunicarse mediante Bluetooth, debe haber otro dispositivo con el que comunicarse. Con Bluetooth, éste puede ser cualquier dispositivo dentro del rango de comunicación. El primer paso para establecer la comunicación es encontrar ese otro dispositivo. El módulo RN4678 dispone de comandos de búsqueda, "F,<número>" que se pueden utilizar para buscar otros dispositivos dentro del alcance. Los comandos "F" envían una señal que es captada por cualquier dispositivo Bluetooth dentro del alcance. Estos dispositivos Bluetooth pueden elegir responder a esta pregunta para hacer saber al dispositivo que pregunta que están ahí. Los dispositivos responden a una consulta enviando su dirección de dispositivo, que puede utilizarse posteriormente para ponerse en contacto con el sistema Bluetooth receptor.

#### Los comandos de investigación

Encontrará información detallada sobre los comandos de consulta en la "Guía del usuario del módulo de modo dual Bluetooth® 4.0 RN4678". Se incluye con los archivos de recursos. Cada vez que se introduzca un nuevo comando, consulte el documento del conjunto de comandos para comprender el nuevo comando. El conocimiento del conjunto de comandos del RN4678 es importante. Los comandos "F,0" o "F,4" hacen que el módulo Bluetooth transmita una señal de consulta a la que pueden responder otros dispositivos Bluetooth. El comando "F,4" devuelve información del dispositivo sin el nombre del mismo. Utilizaremos este comando para reducir la cantidad de información que necesitamos procesar y mostrar. El parámetro opcional <devclass> se utiliza para filtrar qué dispositivos comprobar y se tratará más adelante. Por ahora, se puede utilizar el comando base "F".

Invocando el comando "F,4" recibiremos las siguientes respuestas:

- 123456789012 - La dirección hexadecimal de 12 dígitos del dispositivo Bluetooth que responde a la consulta. Tenga en cuenta que no todos los dispositivos dentro del alcance pueden responder. Los dispositivos Bluetooth pueden configurarse para ignorar el comando de consulta. Esto se tratará en el próximo capítulo.
- Consulta completada - Se recibe cuando la consulta ha finalizado (por defecto 10 segundos) Err - Se ha producido un error

Se pueden recibir varias respuestas: una respuesta de dirección de cada dispositivo que responda a la consulta y un mensaje de confirmación una vez que se hayan recibido todas. Se puede realizar una comprobación y, si no se trata de una respuesta de error, se puede mostrar la dirección.

#### Parámetros adicionales de la consulta

Los comandos de consulta pueden recibir parámetros adicionales opcionales que pueden utilizarse para filtrar las respuestas.

**F,<0-5>,<valor 2>**

Donde <0 - 5> es el índice del modo de consulta

y <valor 2> depende del valor del modo de consulta

Ejemplo: **F,2,001F00** (espera 3 bytes), escaneará el tiempo por defecto (fijado por el comando **SL**) usando el COD 0x1F00

Tenga en cuenta que los índices de modo de consulta 0 a 4 sólo se aplican a Bluetooth Classic y sólo el índice de modo de consulta 5 permite un escaneo BLE

## Ejemplo de descubrimiento

Hay 4 dispositivos Bluetooth dentro del alcance, denominados Blue1-Blue4. Los dispositivos están configurados de la siguiente manera:

Dispositivo	Dirección	Configuración
Azul1	00809872F3D4	Acepta órdenes de consulta
Azul2	00809864DD44	Acepta órdenes de consulta
Azul3	00809894E620	No acepta órdenes de consulta
Azul4	00809894E5D5	Acepta órdenes de consulta

El dispositivo Azul1 transmite el comando de

consulta. Se reciben las siguientes respuestas:

- 00809864DD44
- 00809894E5D5

## Ejercicio 1: Descubrir dispositivos Bluetooth

### Introducción

Para poder comunicarse con un dispositivo Bluetooth es necesario saber que el dispositivo está ahí. Por lo tanto, el primer paso en las comunicaciones Bluetooth es averiguar qué dispositivos están presentes y poder identificarlos para poder establecer comunicaciones con ellos.

### Objetivos

- Desarrolle un programa que realice una búsqueda de dispositivos Bluetooth y muestre las direcciones de los dispositivos encontrados en la pantalla LCD.
- Mostrar un mensaje de finalización una vez recibidas todas las respuestas.

### Requisitos previos

- Conocimiento de los componentes e iconos estándar de Flowcode, como los iconos de decisión, el componente LCD y las macros.
- Comprensión del envío de comandos Bluetooth (consulte el archivo de ayuda del componente Bluetooth en Flowcode para obtener información detallada sobre las macros implicadas).
- Comprensión de la recepción y recuperación de datos de mensajes (consulte el archivo de ayuda del componente Bluetooth para obtener información detallada sobre las macros implicadas y un resumen de las estrategias pertinentes).

### Requisitos de hardware y software

Se necesitan los siguientes elementos de hardware:

- Solución Bluetooth para el programa Ejercicio.
- 1 o más dispositivos Bluetooth adicionales para demostrar el programa. Nota: La segunda placa Bluetooth de la solución Bluetooth puede utilizarse para este Ejercicio. El programa BT\_EX1\_NODE\_A.FCFX puede utilizarse para configurar la placa.
- Configure Flowcode y los componentes de hardware como se especifica en la sección de puesta en marcha.

### Información sobre el ejercicio

El comando de consulta es "F,4", consulte el conjunto de comandos en el documento Guía del usuario del RN4678 para obtener información detallada sobre los comandos del RN4678.

Las respuestas esperadas son:

- Una dirección hexadecimal de 12 dígitos de los dispositivos que responden.
- Un mensaje de error.

"Encontrado x" una vez finalizada la Investigación.

Los objetivos pueden desglosarse de la siguiente manera: Tareas

- Enviar el comando de consulta.
- Compruebe las respuestas.
- Muestra la dirección de cualquier dispositivo que responda.
- Mostrar el mensaje de finalización una vez completada la consulta.

## Resultado del aprendizaje

Los principales resultados de aprendizaje de este ejercicio son

- Comprender los comandos básicos de los dispositivos Bluetooth
- Comprender el proceso de descubrimiento
- Creación y envío de comandos
- Comprobación y recuperación de respuestas

## Comprobación de respuestas específicas Tareas adicionales

Las siguientes son tareas adicionales que pueden implementarse para añadir características extra o mejorar la funcionalidad del programa básico:

- Implemente su propio recuento de cuántas direcciones se han encontrado.
- Realizar una comprobación básica de errores de los valores de retorno del componente Flowcode.
- Compruebe si hay una respuesta de error.
- Recuperar y mostrar el nombre o nombres de los dispositivos utilizando el comando "F,0" en lugar de "F,4".

## Aplicación práctica: Detección de dispositivos Bluetooth Descubrir y ser descubierto

Tenga en cuenta que los programas creados para el Ejercicio 1 deben utilizarse junto con el programa creado en el Ejercicio 2 para formar un par de tableros descubribles y descubridores. Usar los dos ejercicios como un par enlazado permite a los estudiantes explorar el descubrimiento y la descubribilidad usando el mismo programa del Ejercicio 1 en ambos Ejercicios, dándoles continuidad y la posibilidad de ver el Ejercicio anterior en acción.

Los dos programas del Ejercicio 1 y del Ejercicio 2 formarán la parte central de la mayoría de los ejercicios posteriores, añadiéndose comandos adicionales a medida que avance el curso. Como tal, se puede adoptar un enfoque de continuidad en el que los programas finales de los ejercicios anteriores se pueden utilizar a menudo como punto de partida para el siguiente. Este enfoque permite a los estudiantes ver cómo el proceso fluye y crece a medida que avanzan en los ejercicios.

El programa que los estudiantes tienen que escribir en el Ejercicio 1 es para descubrir cualquier otro dispositivo Bluetooth dentro del alcance. Para ello es necesario que el programa esté activo en el dispositivo Bluetooth correspondiente y que dicho dispositivo Bluetooth sea detectable.

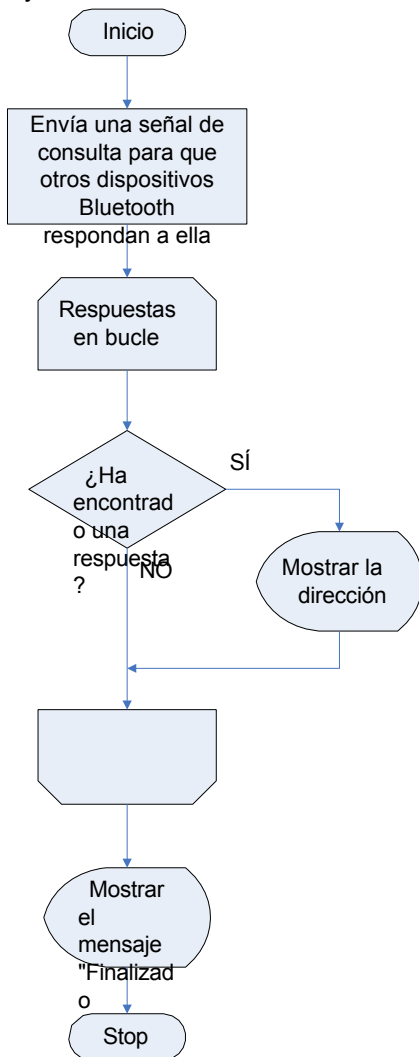
El componente Flowcode BL0170 RN4678 tiene una función Initialise que configura el dispositivo Bluetooth de acuerdo con los ajustes de las propiedades del componente. El nodo A del ejercicio de ejemplo 1 configura el dispositivo como detectable sin necesidad de código adicional en el programa.

Esto permite al alumno empezar con un dispositivo detectable antes de pasar a aprender los comandos del dispositivo.

# Planificación del programa

Antes de escribir un programa es necesario tener una idea de cuáles son los objetivos del mismo y un esquema de cómo se va a llevar a cabo. A continuación se presenta un plano rudimentario para un programa de Investigación.

El plan consiste en enviar la señal de consulta y, a continuación, realizar un bucle para comprobar si hay respuestas y mostrar el anuncio.



de cualquier dispositivo que responda, y el mensaje de finalización una vez completada la consulta.

Los requisitos de hardware para lograr el plan incluyen un módulo Bluetooth y un dispositivo de visualización que sea capaz de mostrar la dirección completa de 12 dígitos de los dispositivos que responden. En este caso utilizaremos la pantalla LCD BL0169.

Los requisitos de software incluyen:

- Envío de una señal de consulta
- Comprobación de las respuestas
- Obtención de los datos de respuesta Visualización de los datos.

## Macros necesarias

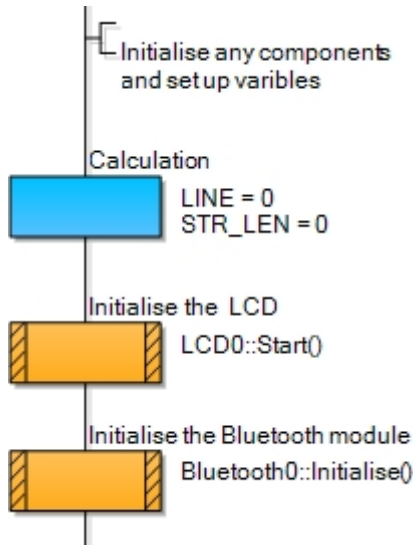
Asumiendo que el estudiante tiene una familiaridad básica con Flowcode y los componentes básicos tales como la pantalla LCD, entonces sólo es necesario cubrir las macros específicas de Bluetooth.

Las siguientes macros de componentes Bluetooth serán necesarias para el programa:

- Inicializar
- EnterCommandMode
- EnviarComando
- RecibirByte



## Inicialización de



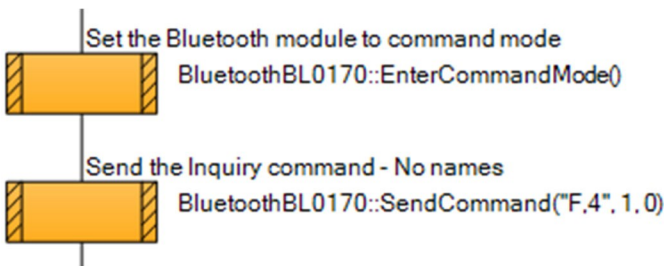
El flujo básico del programa es comenzar inicializando los componentes y estableciendo los valores por defecto de las variables. Añade una macro LCD Start y una macro Bluetooth Initialise.

La macro Inicializar Bluetooth es necesaria para todos los programas que utilicen el componente Bluetooth. El mejor lugar para añadir macros de inicialización es justo al principio del programa.

Añada un icono de cálculo e introduzca los valores por defecto de las variables. Esto puede actualizarse mientras se construye el programa para añadir las nuevas variables que sean necesarias. Establecer valores por defecto es una buena práctica para evitar errores debidos a valores erróneos.

## Enviar una orden

Una vez configurados todos los componentes y variables, se puede plantear la primera tarea, la de enviar la orden de consulta.

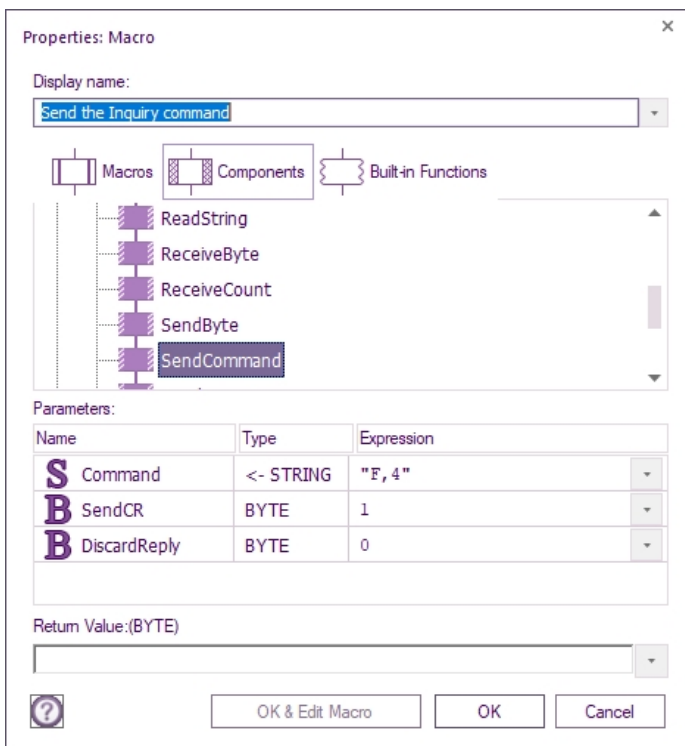


Antes de poder enviar un comando al módulo Bluetooth, es necesario ponerlo en modo comando con la función EnterCommandMode del componente.

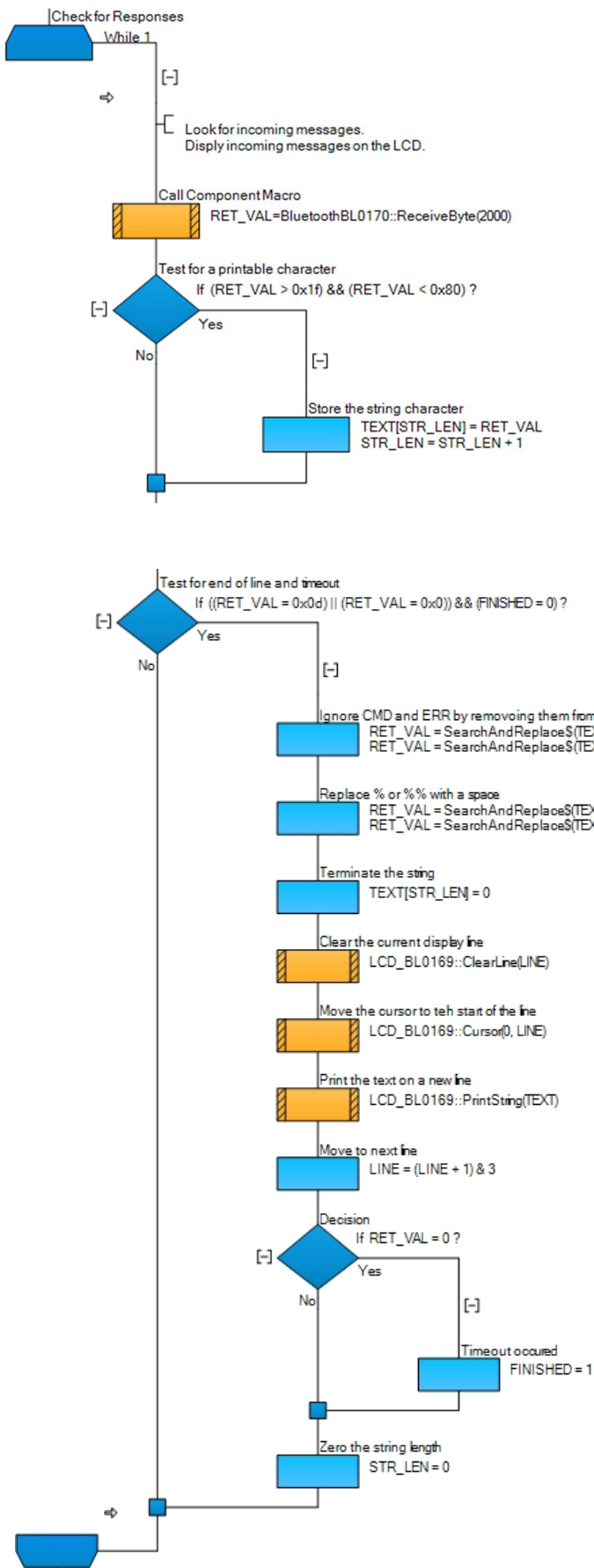
El comando de consulta puede enviarse ahora utilizando la función SendCommand. Introduzca "F,4" como Comando y ajuste SendCR al valor 1 y DiscardReply a 0.

Esta función devuelve un valor de 1 si el comando se envía correctamente. Pero el valor de retorno no se comprobará en este ejemplo para evitar que este primer programa sea demasiado complejo. Sin embargo, en general es una buena práctica utilizar el valor de retorno para la comprobación básica de errores para asegurarse de que los comandos se han enviado.

La pantalla LCD PrintString puede utilizarse para mostrar mensajes de progreso, como "Iniciando consulta".



## Comprobación de las respuestas



Una vez enviada la consulta, el siguiente paso es escuchar las respuestas.

Puede haber varias respuestas, por lo que se necesita una estructura de bucle para pasar por todas ellas, lo que se hace introduciendo un bucle continuo "while 1".

Una llamada a ReceiveByte devolverá caracteres a medida que se reciben del módulo Bluetooth. Colocamos este carácter en la variable RET\_VAL.

A continuación, comprobamos si esta variable es un carácter imprimible (véase una tabla ASCII para más detalles). Si es imprimible, la almacenamos en un búfer de cadena temporal llamado TEXT.

Las cadenas de texto recibidas del módulo Bluetooth suelen terminar (se terminan) con un carácter de retorno de carro, que se representa con el número 0x0d, expresado en notación hexadecimal.

Cuando recibimos este carácter entonces mostramos la cadena recibida que está contenida en nuestro buffer de TEXT.

La función PrintString espera que la cadena de texto termine (terminada) por un carácter de valor NULL o 0, por lo que primero insertamos uno al final de nuestro buffer TEXT.

Si se devuelve un NULL entonces no se accederá a la función de impresión, hasta que se detecte un nuevo carácter imprimible.

También observará en este diagrama de flujo de ejemplo que primero comparamos el búfer de TEXT con las respuestas esperadas que no queremos mostrar, en cuyo caso eliminamos las respuestas no deseadas del búfer de TEXT.

La variable STR\_LEN lleva la cuenta de la longitud del texto entrante, así que restablece el valor a 0 antes de volver a la función

## Ejecución del programa completo FINALIZADO

El programa completo se encuentra en el ejemplo BT\_EX1\_NODE\_B.FCFX.

Este programa se descarga en la placa del procesador de destino del ejercicio con una placa BL0170 Bluetooth conectada al puerto B en el caso del procesador BL0011 PIC.

Además, al menos otro dispositivo Bluetooth debe estar activo y detectable. Puede ser la otra placa de la solución Blue-tooth o un dispositivo Bluetooth de terceros.

Sin embargo, tenga en cuenta que no todos los dispositivos Bluetooth responderán. Al demostrar el programa puede ser prudente probar los dispositivos Bluetooth antes de usarlos para la demostración para comprobar si pueden ser descubiertos o no.

Por ejemplo, si tienes un teléfono móvil con Bluetooth, tendrás que activar su Bluetooth y configurar la visibilidad del teléfono para que no quede oculto.

Compile y descargue el programa en el microcontrolador de una de las placas Bluetooth. Desenchufa brevemente el cable de alimentación, vuelve a enchufarlo y pulsa el interruptor Reset de la placa programadora. Esto asegura que el módulo Bluetooth se reinicie correctamente.

La pantalla LCD mostrará mensajes de progreso y la dirección de 12 dígitos de los dispositivos encontrados, terminando con un mensaje de finalización del número de dispositivos encontrados.

# Descubribilidad

## Teoría: Descubribilidad

El comando de consulta descrito en el capítulo anterior se envía a todos los dispositivos Bluetooth. Sin embargo, no todos los dispositivos responderán a la señal de consulta. Esto se debe a que los dispositivos Bluetooth pueden ocultarse de otros dispositivos.

Los dispositivos también pueden configurarse para ser detectables, en cuyo caso responderán al comando Inquiry.

## Configuración para la puesta en marcha

Los dispositivos Bluetooth tienen una serie de opciones de configuración que abarcan desde las claves de paso y la encriptación hasta la velocidad en baudios y el número de timbres antes de responder. Por lo general, se utilizan los valores predeterminados para que los dispositivos funcionen "nada más sacarlos de la caja". Sin embargo, es una buena práctica configurar los dispositivos con los ajustes correctos al encenderlos para asegurarse de que funcionan como se desea.

En el módulo RN4678, estas opciones se pueden configurar con los distintos comandos, detallados en la Guía del usuario del módulo de modo dual Bluetooth® 4.0 RN4678. Estos comandos se pueden agrupar en Comandos Set, Comandos Get y Comandos Action.

## Uso de los comandos Set

Los comandos de configuración cambian las configuraciones del RN4678 y surten efecto después de reiniciar mediante el comando R,1, un reinicio completo o un ciclo de alimentación. Algunos comandos de configuración sólo se aplican al funcionamiento Bluetooth Classic o BLE. Los comandos de configuración tienen parámetros e incluyen cosas como:

- Modo de funcionamiento Bluetooth
- Método de autenticación
- Clase e ID del dispositivo
- Ventana de consulta
- Nombre del dispositivo
- Código PIN de seguridad
- Fiabilidad del flujo de datos
- Direcciones remotas
- Nombre del servicio
- Velocidad de transmisión UART
- Activar o desactivar el símbolo del sistema

Todos los cambios de configuración realizados mediante comandos Set permanecen en la memoria no volátil (NVM) y sobreviven al ciclo de encendido. Cualquier cambio de configuración debe tener efecto después de un reinicio.

## Uso de los comandos Get

Los comandos Get recuperan y muestran la información almacenada del dispositivo. La mayoría de estos comandos no tienen parámetros adicionales. Estos comandos devuelven valores de ajuste e información sobre el estado del dispositivo.

## Uso de comandos de acción

Los comandos de acción realizan acciones como escaneos de consulta, conexión y entrada o salida del modo Comando. Algunos comandos de acción solo se aplican a Bluetooth Classic o BLE.

Todos los comandos de acción surten efecto inmediatamente, pero no tienen efecto después de la desconexión de la alimentación.

## Hacer que un dispositivo sea detectable

Para este ejercicio nos interesa que el dispositivo responda automáticamente y sea detectable y conectable, y que podamos enviarle datos.

Para ello, la secuencia de eventos y la configuración de los comandos son las siguientes:

### Entrar en modo comando

El componente Flowcode RN4678 tiene una función que implementa esto llamada "EnterCommandMode". Esto envía la secuencia requerida de \$\$\$ caracteres al dispositivo.

### Esperar a que se ejecute el comando

Tras cada solicitud de comando debemos esperar la confirmación y disposición del dispositivo para aceptar un nuevo comando. El dispositivo solicita un comando con la cadena de texto CMD> . Así que usamos la función RN4678 de Flowcode WaitForStringValue("CMD> ") para esperar hasta que el dispositivo esté listo para un nuevo comando.

### Velocidad en baudios y ajustes

Existe un comando, SU,<Índice de Velocidad de Baudios>, que establece la velocidad de baudios del comando serie. Sin embargo, está ajustada de fábrica a 115.200, por lo que debe dejarse en este valor por defecto para los ejercicios de este curso.

### Modo de autenticación

Para los ejercicios iniciales configuramos el modo de autenticación como "Sólo funciona". Este modo funciona sin necesidad de mostrar o introducir ningún pin de seguridad. El comando es **SA,2**

### Modo esclavo

Cuando se ejecutan algunos comandos del modo "Salve", el Bluetooth activa automáticamente el modo "Salve" .

### Activar la configuración actualizada

Ahora que hemos utilizado los comandos Set para cambiar los parámetros, debemos reiniciar el dispositivo. Cualquier cambio en la configuración del dispositivo mediante los comandos Set no surtirá efecto hasta que se reinicie el dispositivo. Esto se hace con el comando de acción **R,1**.

### Espere a que se reinicie y salga del modo de comandos

Tras el reinicio salimos del modo comando utilizando la función RN4678 de Flowcode LeaveCommandMode.

### Resumen de los comandos básicos de configuración

Un simple conjunto de comandos para hacer el dispositivo detectable son:

Comando	Descripción
SA,2	Modo Secure Simple Pairing (SSP) "Just Works". Este modo funciona sin necesidad de mostrar o introducir ningún pin de seguridad.
R,1	Reinicie el dispositivo para actuar la nueva configuración.

### Introducción

- Para que un dispositivo sea encontrado debe ser detectable. Un dispositivo Bluetooth detectable puede responder a una orden de consulta de otro dispositivo Bluetooth.

### Objetivos

- Desarrolle un programa que configure el dispositivo Bluetooth para que sea detectable. Este es el ejercicio complementario al Ejercicio 1.

### Requisitos previos

- Comprensión del proceso de Descubrimiento tal y como se detalla en el Ejercicio 1. En particular, la capacidad de reconocer una orden de investigación.
- Comprensión de la recepción y recuperación de datos de mensajes (consulte el archivo de ayuda del componente Bluetooth para obtener información detallada sobre las macros implicadas y un resumen de las estrategias pertinentes).
- Requisitos de hardware y software
- Se necesitan los siguientes elementos de hardware:
- Solución Bluetooth para el programa Ejercicio.
- 1 o más dispositivos Bluetooth adicionales para demostrar el programa. Nota: La segunda placa Bluetooth de la solución Bluetooth puede utilizarse para este Ejercicio. El programa del Ejercicio 1 puede usarse para enviar el comando Inquiry, Información del Ejercicio

Los objetivos pueden desglosarse en los siguientes:

- Tareas
- Envía los comandos de configuración.
- Comprueba si hay mensajes.
- Mostrar mensajes.
- Estructuras de control:
- Comprobación y visualización de respuestas.
- Resultado del aprendizaje
- Los principales resultados de aprendizaje de este ejercicio son
- Comprender el proceso de descubrimiento.
- Comprensión de cómo configurar el dispositivo RN4678.
- Conocimiento de los comandos Set y Action.
- Comprobación y recuperación de respuestas.
- Tareas adicionales
- Consulte la configuración de comandos del manual del módulo Bluetooth y haga que el módulo Bluetooth no sea detectable.
- ¿Qué implicaciones tiene esto para la persona que utilice el equipo después de usted?
- Utilizando los programas de los Ejercicios 1 y 2 anota las direcciones Bluetooth de cada uno de los módulos Bluetooth - las necesitarás en el siguiente ejercicio.

## Aplicación práctica: Descubribilidad

El Ejercicio 2 acompaña al Ejercicio 1 en el sentido de que, habiendo mostrado cómo funcionan los dispositivos Descubribles en el Ejercicio 1, el siguiente paso es mostrar cómo hacer que los dispositivos sean descubribles. Además, se discuten los métodos de configuración y el uso de los comandos Set y Action.

Es importante destacar que los comandos utilizados aquí son particulares del módulo específico utilizado (el dispositivo Microchip RN4678), por lo que son características del chip RN4678 y no una parte inherente de Bluetooth. Diferentes chips pueden tener diferentes estructuras de comandos y registros, o incluso no utilizar ninguno. Sin embargo, los principios generales son los mismos.

Antes de empezar a conectar el cable USB al segundo sistema Bluetooth - puedes dejar el programa del Ejercicio 1 en el primer sistema Bluetooth y lo usaremos para probar si el programa creado aquí funciona.

### Desarrollo continuo

Los programas de Estudiantes creados para el Ejercicio 1 se pueden utilizar junto con el programa creado en este ejercicio para formar un par de tableros descubribles y descubridores.

Los dos programas del Ejercicio 1 y del Ejercicio 2 formarán la parte central de la mayoría de los ejercicios posteriores, añadiéndose comandos adicionales a medida que avance el curso. Como tal, se puede adoptar un enfoque de continuidad en el que el programa final de los ejercicios anteriores se puede utilizar a menudo como punto de partida para el siguiente.

## Configurar el dispositivo Bluetooth

Para este Ejercicio el dispositivo Bluetooth necesita ser configurado para ser detectable y conectable. No nos conectaremos a él en este Ejercicio en particular; sin embargo, podemos establecer esta configuración básica ahora.

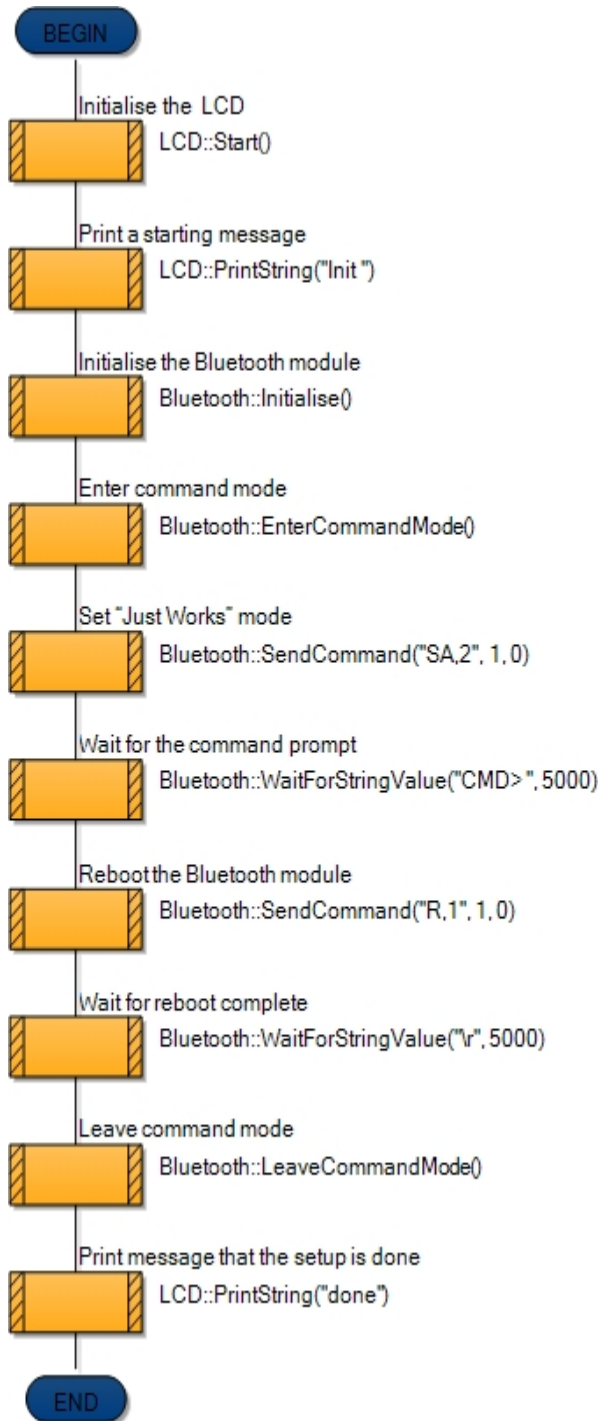
Para este ejercicio utilice los siguientes comandos de configuración:

SA,2

R,1

Estos comandos configurarán el dispositivo para que responda inmediatamente y para que sea detectable y conectable.

## Creación del programa



Las macros *SendCommand* y *WaitFor-StringValue*, detalladas anteriormente, se utilizan para configurar el módulo Bluetooth como en el ejemplo que se muestra aquí.



## Ejecución y demostración del programa

El programa construido aquí puede ser utilizado con el programa de Consulta construido en el Ejercicio 1 Nodo B.

El programa Nodo B del Ejercicio 1 puede usarse en una placa Bluetooth, y este programa del Ejercicio 2 puede usarse como Nodo A en una segunda placa Bluetooth.

- Cree el programa, anotando mientras lo hace la dirección de dispositivo de 12 dígitos de la placa Bluetooth que se utilizará con el programa del Ejercicio 2.
- Descarga y ejecuta el Ejercicio 2 en la placa Bluetooth cuya dirección has anotado.
- Descargue y ejecute el programa Inquiry creado en el Ejercicio 1 en la otra placa.
- Monitoree la pantalla LCD de la tarjeta a descubrir. Cuando se envíe el comando Inquiry se mostrará en la pantalla LCD.

Monitoree la pantalla LCD de la tarjeta del programa de consulta. Cuando la tarjeta detectable reciba el comando de consulta, responderá devolviendo su dirección de 12 bits (el número que anotó anteriormente). Compruebe en la pantalla LCD que se muestra la dirección.

# Conexión de dispositivos Bluetooth

## Teoría: Conectar - Direcciones

Al no existir conexión física entre dos dispositivos, todas las comunicaciones pueden ser captadas por cualquier otro dispositivo Bluetooth. Un problema inmediato es cómo comunicarse con un dispositivo concreto y no hacerlo inadvertidamente con otros.

Todos los dispositivos Bluetooth Classic tienen una dirección (dirección MAC), un número hexadecimal único de 12 dígitos. Esta dirección es el mismo número que se devuelve en respuesta a un comando de consulta. Esta dirección se puede recuperar mediante un comando de consulta o se puede almacenar en el programa si se va a utilizar una dirección de dispositivo específica y se conoce de antemano.

Si un dispositivo Bluetooth es conectable y se conoce su dirección, otros dispositivos Bluetooth pueden iniciar la conexión con él. Por lo general, los dispositivos serán detectables y conectables, de modo que la dirección pueda determinarse a partir de un comando de consulta. Sin embargo, es posible que los dispositivos sean conectables pero no detectables. Por ejemplo, un conjunto de teléfonos que están diseñados para comunicarse sólo entre sí puede estar configurado para ser conectable pero no detectable. El par de teléfonos tiene la dirección de cada teléfono en la memoria, por lo que no se necesita un comando de descubrimiento, y el teléfono está listo para comenzar a comunicarse con su dispositivo asociado.

El comando básico de conexión es:

```
C,<bt_addr>
```

Donde <bt\_addr> es la dirección hexadecimal de 12 dígitos del dispositivo con el que iniciar la conexión.

La dirección MAC de Bluetooth Classic de los módulos RN4678 viene configurada de fábrica y no puede modificarse.

### Introducción

Si se conoce la dirección de un dispositivo Bluetooth, y el dispositivo ha sido configurado para ser conectable, entonces se puede conectar a ese dispositivo.

### Objetivos

- Desarrolla un programa en un sistema Bluetooth (nodo B) que se conecte a otro sistema Bluetooth (nodo A), y muestra la respuesta para mostrar que la conexión ha tenido éxito.

Desarrolle un programa en un sistema Bluetooth (nodo A) que permita a otro sistema Bluetooth (nodo B) conectarse a él. Este programa se suministra en BT\_EX3\_Node\_A.fcfx.

### Requisitos previos

Comprensión del proceso de Descubrimiento, tal y como se detalla en los ejercicios 1 y 2.

### Requisitos de hardware y software

Se necesitan los siguientes elementos de hardware:

- Solución Bluetooth para el programa Ejercicio.

1 o más dispositivos Bluetooth adicionales para demostrar el programa. Nota: La segunda placa Bluetooth de la solución Bluetooth puede ser utilizada para este Ejercicio. Descarga el programa BT\_EX3\_Node\_A.fcfx en este segundo sistema.

### Información sobre el ejercicio

El comando Iniciar Conexión es:

Comando	Descripción
C,<bt_addr>	Donde <bt_addr> es la dirección de la segunda placa Bluetooth, nodo A.

Configure dos programas: uno en el nodo B para conectarse a un segundo dispositivo Bluetooth en el nodo A. Cuando emita el comando de conexión, el nodo receptor enviará el acuse de recibo. Visualice todos los mensajes en la pantalla LCD de los nodos.

### Resultado del aprendizaje

Los principales resultados de aprendizaje de este ejercicio son

Conexión a otro dispositivo Bluetooth.

### Otros trabajos

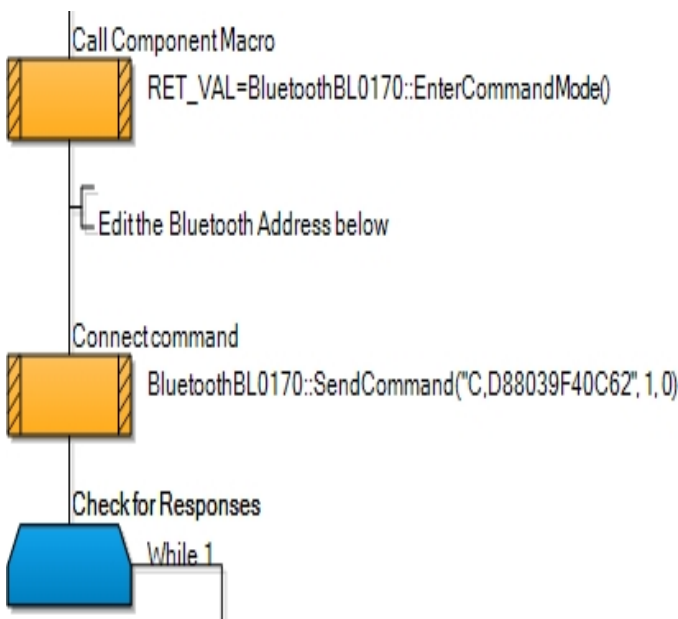
La pantalla LCD sólo tiene 20 caracteres por línea. Diseñe una extensión del código en BT\_EX3\_Node\_A.FCFX para que cualquier desbordamiento de texto se muestre en la siguiente línea de la pantalla.

Al igual que en los ejercicios anteriores, los estudiantes deben desarrollar dos programas: uno en el dispositivo Bluetooth que inicializa la comunicación y otro en el dispositivo que recibe la comunicación inicial.

Este ejercicio es rápido y sencillo desde el punto de vista de la carga de trabajo, pero tiene una serie de escollos técnicos que hay que tener en cuenta.

1) Dirección del dispositivo. El programa de ejemplo BT\_EX3\_Node\_B.FCFX utiliza la dirección de un dispositivo Bluetooth utilizado para pruebas aquí en Matrix. Esto tendrá que ser editado para que coincida con la dirección de la placa que se va a utilizar con el programa de ejemplo. Para ello, debe utilizar la dirección que encontró en el Ejercicio 2 e introducirla como parámetro en el comando

*SendCommand* como se indica a continuación.



Ten claro cuál es el nodo emisor y cuál el receptor y cuáles son sus direcciones.

2) El dispositivo Bluetooth al que se va a conectar necesita haber sido configurado para ser conectable. Una vez más, el programa BT\_EX3\_Node\_A.FCFX se ha configurado teniendo en cuenta esta cuestión.

3) Una vez que se ha establecido una conexión, necesitamos alguna forma de verificar que la conexión se ha realizado. El programa BT\_EX3\_Node\_A.FCFX está configurado para mostrar cualquier mensaje enviado, por lo que podemos simplemente enviar un mensaje y comprobar que es recibido por el otro dispositivo. El programa que escribas debe mostrar el mensaje devuelto por el nodo A.

Actualmente no se realiza ninguna comprobación de errores. El programa asume que la conexión tendrá éxito, lo que puede no ser siempre el caso. Un método mejor sería utilizar respuestas para comprobar errores en el proceso. Las respuestas y la comprobación de errores se tratarán en un capítulo posterior.

## Reiniciar los sistemas

El módulo Bluetooth no está equipado con un botón de reinicio. El único mecanismo de reinicio es desconectar la alimentación. Si ha programado el módulo Bluetooth para llevar a cabo alguna actividad, pulsar el botón de reinicio en la placa del procesador no reiniciará necesariamente el módulo Bluetooth.

Por esta razón, al desarrollar pares de programas, **puede** ser necesario quitar la alimentación del sistema y reiniciar los módulos Bluetooth.

## Teoría: Llaves de paso y conexión

Bluetooth se comunica por radio, un medio de transmisión intrínsecamente inseguro. Cualquier dispositivo dentro del alcance de la misión puede recibir las señales enviadas a cualquier otro dispositivo. El uso de una dirección única resuelve el problema de especificar con qué dispositivo quieres comunicarte, pero se necesitan otros métodos para evitar el acceso no autorizado a un dispositivo.

El sistema de seguridad básico utilizado con Bluetooth se denomina emparejamiento. El emparejamiento se produce cuando dos dispositivos se conectan entre sí utilizando tanto una dirección de dispositivo como la "clave de enlace" secreta de los dispositivos, conocida como Passkey. La clave no se puede recuperar de otro dispositivo de la misma forma que la dirección. Para poder conectarse a un dispositivo, debe conocer el número de Passkey de ese dispositivo.

### Envío del comando Passkey

Para configurar un dispositivo de modo que se pueda emparejar con una Passkey es necesario configurar el dispositivo. El comando de configuración para configurar una Passkey es:

```
SP,<clave de acceso>
```

Donde <passkey> es el número PIN de Passkey. La Passkey es un número de 4 dígitos (para el emparejamiento de código PIN heredado, se utiliza un PIN de 6 dígitos para la autenticación SSP en BLE). Por lo general, los números PIN de Passkey se suministran con la documentación que acompaña a un dispositivo concreto. Los números PIN de los dispositivos deben guardarse en un lugar seguro, igual que se haría con los números PIN de las tarjetas bancarias.

### Iniciar el emparejamiento y la conexión

Una vez definida la clave de acceso, la conexión de datos puede realizarse iniciando el comando Conexión del siguiente modo:

```
C,<bt_addr>
```

Donde <bt\_addr> es la dirección del dispositivo Bluetooth Classic al que se va a conectar.

En muchos casos, como cuando se empareja un teléfono móvil con unos auriculares, la Passkey estará en la documentación del producto y el programa le pedirá que la introduzca.

## Ejercicio 4: Claves de acceso y conexión

### Introducción

Los dispositivos pueden emparejarse entre sí para comunicarse. El emparejamiento requiere la dirección del dispositivo con el que se quiere emparejar, y un valor de Passkey para establecer el contacto. Aquí tendrás que desarrollar dos programas: uno que establezca cuál es la clave de paso en un sistema, y otro que se conecte a ese sistema y le envíe datos.

### Objetivos

- Desarrolla programas para dos sistemas Bluetooth que permitan el emparejamiento. Desarrollar un programa para el Nodo A que asigne una Passkey de "1234", y hacerlo detectable. Desarrolle rutinas que muestren en la pantalla LCD cualquier dato enviado al nodo. Desarrolle un programa para el nodo B que se empareje con el nodo A usando la dirección y la Passkey del nodo A.
- Una vez establecida la conexión, se establece una comunicación sencilla entre los sistemas que muestra que se está produciendo la comunicación, es decir, un contador en el nodo emisor cuenta los datos como un único byte que también se muestra en el nodo receptor.

### Requisitos previos

- Comprensión del proceso de conexión detallado en el Ejercicio 3.

Comprensión de la creación, envío y recepción de comandos, tal y como se detalla en los ejercicios 1 y 2.

### Requisitos de hardware y software

Se necesitan los siguientes elementos de hardware:

- Ambas soluciones Bluetooth son necesarias para este Ejercicio.
- Solución 1, nodo A, se conectará a.

Solución 2, nodo B, iniciará la conexión.

### Información sobre el ejercicio

Comando	Descripción
SP,<clave de acceso>	Donde <clave de acceso> es el valor de la clave de acceso del dispositivo con el que se va a emparejar.
C,<bt_addr>	Donde <bt_addr> es la dirección de la segunda placa Bluetooth. Si el comando de conexión tiene éxito, se devolverá una respuesta CONNECT <bd_addr>.

Una vez que se ha establecido la conexión, haga un bucle y envíe los números del 0 al 9 para que se muestren en el otro dispositivo.

### Resultado del aprendizaje

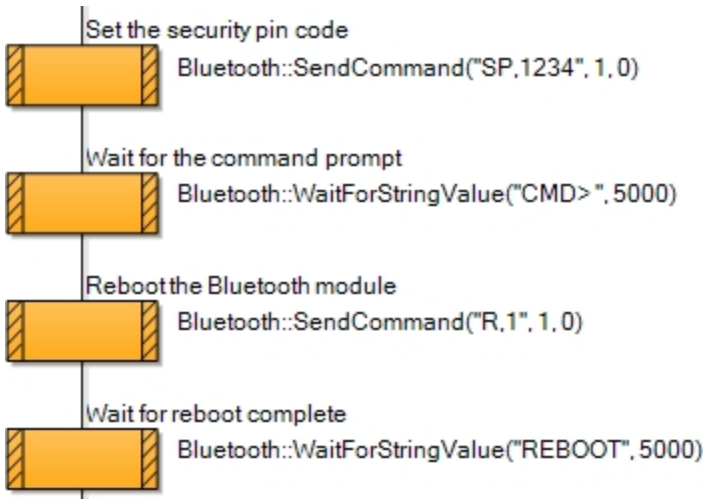
- Comprensión del proceso de conexión.

Comprensión de la función y las implicaciones de seguridad de la Passkey.

### Otros trabajos

- Utilizando un teclado y una variable de matriz, desarrolle dos programas que permitan establecer la clave de acceso del sistema receptor y la clave de acceso del sistema transmisor.

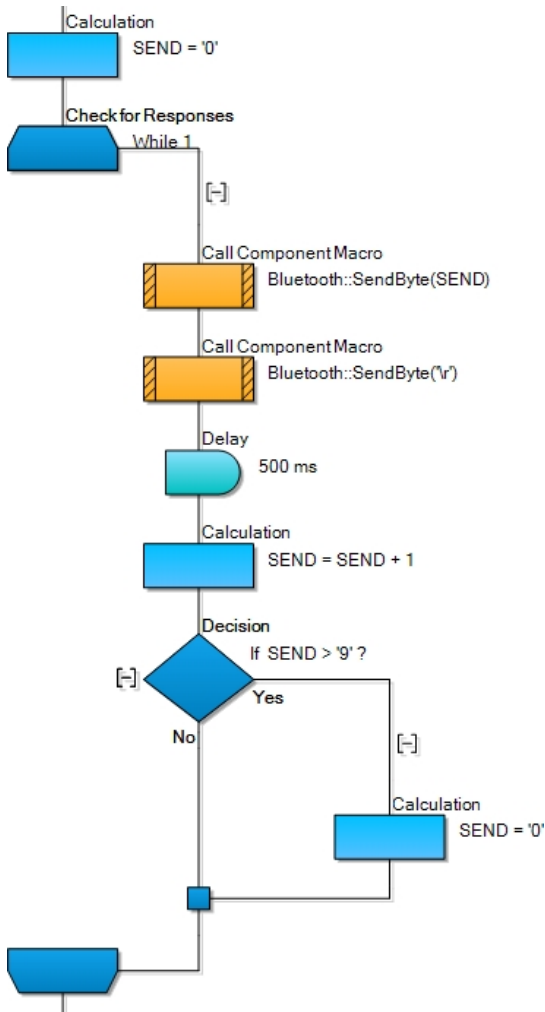
## El programa básico



El emparejamiento es un proceso clave en las operaciones Bluetooth. La clave de acceso debe configurarse y, como ocurre con todos los comandos "set", se activa después de reiniciar el dispositivo, lo que puede hacerse con el comando "R,1".

Si los datos se conocen de antemano, basta con introducirlos en el programa, crear los comandos y enviarlos. Si la clave de acceso y la dirección no se conocen de antemano, habrá que establecerlas.

La dirección del dispositivo Bluetooth puede obtenerse descargando el programa BT\_TEST Flowcode. Si lo desea, puede anotar las direcciones de las distintas tarjetas.



Para demostrar que la conexión funciona, se puede enviar una simple señal de repetición 0-9 que se mostrará en el dispositivo receptor.

Las direcciones de los dispositivos se pueden recuperar utilizando el comando Inquiry. Sin embargo, saber si necesitamos conectar el dispositivo 007284972989 o 00234869030 es otra cuestión. El comando Inquiry puede ampliarse para obtener información adicional además de la dirección de 12 dígitos. El comando "F,0" no sólo recupera la dirección del dispositivo, sino también su nombre descriptivo. El nombre amigable es una cadena de descripción del dispositivo que es más fácil de entender para los usuarios que la dirección del dispositivo.

Por ejemplo:

012345678912, "NodoA"

012345678913, "NodoB"

Mostrar la dirección amigable puede ayudar a los usuarios a seleccionar el dispositivo correcto de una lista. La práctica de mostrar el nombre amigable es una simple modificación al código de respuesta de pantalla del Ejemplo 1. Añadiendo en el espacio de comandos los primeros 14 caracteres pueden ser mostrados en la línea superior del LCD, y el resto en la línea inferior moviendo el cursor cuando los primeros 14 caracteres han sido mostrados.

Además, podemos añadir los 12 caracteres de dirección en una matriz, almacenando la dirección actual para su uso posterior. Esto puede hacerse al mismo tiempo que imprimimos los caracteres. Añade los 12 primeros caracteres a la dirección actual.

Obtener la clave de acceso es más difícil, ya que no se puede recuperar mediante un comando. En los ejemplos utilizados en este curso se utiliza la clave "1234". En la práctica, puede ser necesario obtener la clave de acceso de la documentación del dispositivo e introducirla en el código del programa, o introducirla en el programa manualmente.

## Reiniciar los sistemas

La placa Bluetooth BL0170 no está equipada con un botón de reinicio. El único mecanismo de emitir un reinicio es volver a encenderla o emitir el comando de reinicio (R,1). Si ha programado el módulo Bluetooth para realizar alguna actividad, pulsar el botón de reinicio en la placa programadora no reiniciará necesariamente el módulo Bluetooth.

Por esta razón, al desarrollar pares de programas puede ser necesario desconectar la alimentación del sistema y reiniciar los módulos Bluetooth.

Para este ejercicio te recomendamos que desconectes la alimentación de ambos sistemas cada vez que descargues un programa. A continuación, enciende el sistema receptor, pulsa reset y déjalo unos segundos para que se configure. A continuación, encienda el sistema de transmisión, pulse reset y espere unos segundos para que se configure.



## Comprobación de las respuestas

### Teoría: Comprobación de las respuestas

En algunos de los ejercicios anteriores enviamos configuraciones o comandos y asumimos que se llevarán a cabo. Un método más útil sería monitorizar el resultado de la macro del componente Flowcode y comprobar adicionalmente la respuesta del proceso.

La mayoría de las macros de componentes Bluetooth tienen un valor de retorno que indica si la petición ha tenido éxito o no. Este valor de retorno puede ser probado y utilizado para determinar qué se debe hacer a continuación, ya sea continuar con el siguiente paso del programa o saltar al final y abortar.

Cuando se envía un comando al dispositivo Bluetooth, generalmente se envía una respuesta a ese comando. Para muchos comandos y ocasiones la respuesta será un mensaje 'OK'. Sin embargo, muchos comandos tienen respuestas específicas. El comando Descubrir, por ejemplo, envía direcciones como respuesta, junto con un mensaje de finalización del proceso.

### Respuestas solicitadas y no solicitadas

Cuando una respuesta es el resultado directo del envío de un comando, se trata de una respuesta solicitada, es decir, una respuesta que se espera. Por ejemplo, enviar un comando SP,1234 dará como resultado una respuesta solicitada de "OK".

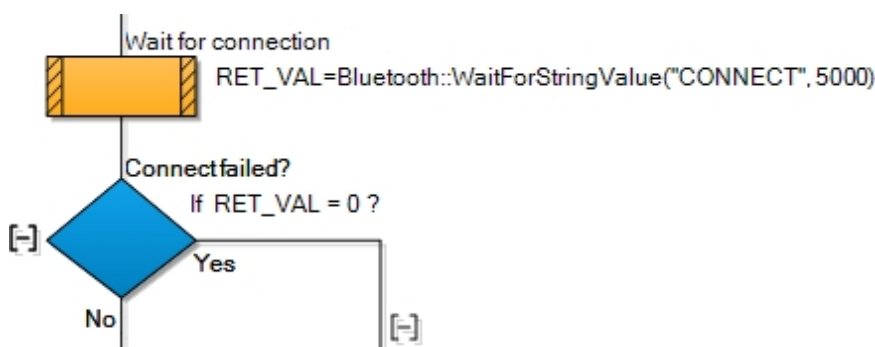
A veces se envía una respuesta que no responde a un comando específico. Se trata de una respuesta "no solicitada". Las respuestas no solicitadas pueden enviarse en distintos momentos para indicar la necesidad de una mayor interacción o para proporcionar información adicional. Por ejemplo, las respuestas CONNECT enviadas durante el establecimiento de las comunicaciones para informar al otro dispositivo del estado actual de la operación.

### Macros de tratamiento de respuestas

Hay algunas macros de componentes Bluetooth disponibles para ayudar con el manejo de las respuestas:

#### **WaitForStringValue**

Esto detiene el flujo del programa hasta que se reciba un mensaje de respuesta específico o expire el tiempo de espera. Por ejemplo, aquí esperamos el mensaje "CONNECT" durante un máximo de 5 segundos (5.000 mS).



Estas macros de componentes nos permiten leer y, por tanto, probar o mostrar cualquier texto que se reciba del módulo Blue- tooth.

Por ejemplo, podemos crear un búfer de texto global MESSAGE y utilizar la macro *ReadString* para leer la respuesta en este búfer. *ReadString* tiene como parámetro el tamaño de la memoria intermedia (32 bytes en este ejemplo).

The screenshot shows a development environment with two main panels. On the left is the 'Project Explorer' and 'Locals' panel, and on the right is a flowchart titled 'Display Responses'.

**Project Explorer / Locals [DISP\_RESP]**

Expression	Value
<b>Globals</b>	
Constants	
Variables	
<b>B</b> CHAR	n/a
<b>B</b> CNT	n/a
<b>B</b> ERROR_VAL	n/a
<b>B</b> FINISHED	n/a
<b>B</b> INDEX	n/a
<b>S</b> MESSAGE[32]	n/a
<b>B</b> RET_VAL	n/a
<b>B</b> STR_LEN	n/a
<b>Locals [DISP_RESP]</b>	
Parameters	

**Flowchart: Display Responses**

```

    graph TD
      BEGIN([BEGIN]) --> Look[Look for incoming messages.  
Display incoming messages on the LCD.]
      Look --> Go[Go to next line  
LCD_BL0169::Cursor(0, 1)]
      Go --> Read[Call Component Macro  
MESSAGE=BluetoothBL0170::ReadString(32)]
      Read --> Print[Call Component Macro  
LCD_BL0169::PrintString(MESSAGE)]
      Print --> END([END])
    
```

## Ejercicio 5: Comprobación de las respuestas

### Introducción

Cuando el dispositivo Bluetooth recibe un comando, devuelve una respuesta. Comprobando estas respuestas es posible tanto controlar cualquier error que se produzca como esperar las respuestas esperadas antes de continuar con una secuencia de comandos.

### Objetivos

- Desarrolle programas para dos sistemas Bluetooth que permitan la conexión. Desarrolle un programa para el Nodo A que asigne una Passkey de "1234", y hacerlo detectable. Desarrolle rutinas que muestren cualquier dato enviado al nodo en la pantalla LCD. Puedes usar el programa del Ejercicio 4 para esto.
- Desarrolle un programa para el nodo B que se conecte con el nodo A utilizando la dirección del nodo A.

### Informar de los errores que se

### produzcan en el nodo B

### Requisitos

- Comprensión del proceso de conexión detallado en el Ejercicio 4.

Comprensión de la macro WaitForStringValue tal y como se detalla en la página anterior y en la ayuda del componente Bluetooth.

### Requisitos de hardware y software

Se necesitan los siguientes elementos de hardware:

- Para este ejercicio se necesitan los dos conjuntos Bluetooth.
- Conjunto 1, nodo A, esperará conexión. El conjunto 2, nodo B, iniciará la conexión.

### Información sobre el ejercicio

Comando	Descripción
SP,<clave de acceso>	Donde <clave de acceso> es el valor de la clave de acceso del dispositivo con el que se va a emparejar.
C,<bt_addr>	Donde <bt_addr> es la dirección de la segunda placa Bluetooth. Si el comando de conexión tiene éxito, se devolverá una respuesta CONNECT <bd_addr>.

### Resultado del aprendizaje

Los principales resultados de aprendizaje de este ejercicio son

- Secuencias de comandos y respuestas.
- Tipos de respuesta.
- Comprobación de errores y control de secuencias.
- Respuestas no solicitadas.

## Aplicación práctica: Comprobación de las respuestas

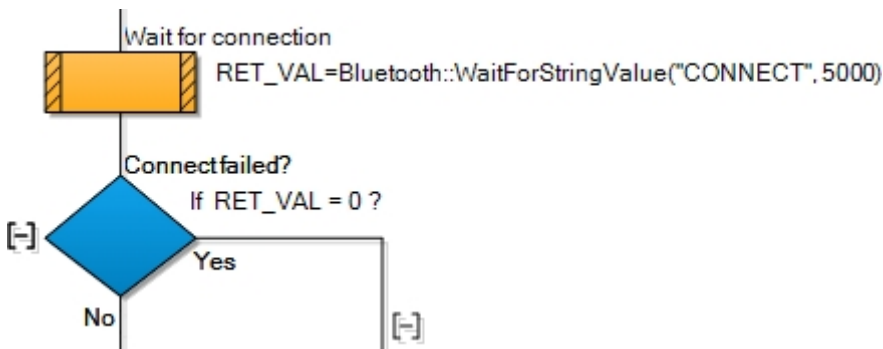
Dos datos importantes a tener en cuenta son las direcciones de los dispositivos Bluetooth y las claves. Para este ejercicio se asume que se utilizará la clave general "1234". Sin embargo, los dispositivos Bluetooth deberán modificarse en consecuencia.

La tarea requiere dos programas, uno para iniciar la conexión y otro para mostrar los datos enviados al dispositivo emparejado. Los programas desarrollados en el Ejercicio 4 pueden utilizarse como base del Ejercicio 5 permitiendo al alumno ver cómo progresa su código.

El programa de visualización sólo necesita mostrar los datos enviados, por lo que el programa de visualización del Ejercicio 4 puede utilizarse tal cual. El objetivo principal será ampliar el programa para incluir la comprobación de la respuesta.

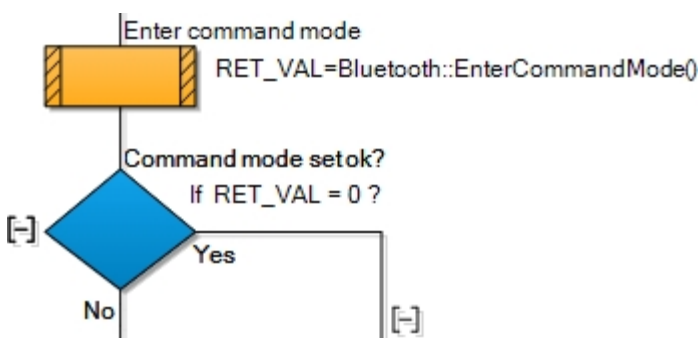
### Uso de la macro WaitForStringValue

Esto detiene el flujo del programa hasta que se reciba un mensaje de respuesta específico o expire el tiempo de espera. Por ejemplo, aquí esperamos el mensaje "CONNECT" durante un máximo de 5 segundos (5.000 mS).



### Metodología de comprobación de errores

Debe adoptarse una metodología básica de comprobación de errores en la que se compruebe si hay errores en los comandos. Un fallo indica que hay un problema al intentar enviar la orden o al completarla.



Por ejemplo, aquí probamos el valor de retorno para comprobar que se ha entrado con éxito en el modo comando.

Una vez que se ha enviado una orden, se puede utilizar *WaitForStringValue* para comprobar si se ha recibido la respuesta esperada.

# Introducción a Bluetooth Low Energy (BLE)

## 10.1 Modos de funcionamiento de Bluetooth Low Energy (BLE)

El módulo Microchip RN4678 Bluetooth Dual Mode utilizado en este curso de formación es compatible con la comunicación Bluetooth Classic y Bluetooth Low Energy (BLE).

En Bluetooth Classic, RN4678 implementa el perfil de puerto serie (SPP) estándar que admite la transferencia de datos serie entre dos dispositivos Bluetooth Classic.

Además de SPP para la conectividad Bluetooth Classic, el RN4678 dispone de un servicio privado de perfil de atributo genérico (GATT) para la transferencia de datos en serie entre dos dispositivos BLE.

Este servicio de transmisión de datos BLE proporcionado en el RN4678 se denomina "UART transparente". Por lo tanto, el RN4678 es un módulo Bluetooth de modo dual, que admite la conectividad de datos serie Bluetooth Classic y BLE.

El módulo funciona en dos modos: Modo Datos (por defecto) y Modo Comandos.

Cuando el módulo está "conectado" a otro dispositivo y en modo Datos, actúa como una tubería de datos.

Es decir, todo lo que se recibe de UART se pasa al dispositivo par conectado a través de SPP si está conectado a un dispositivo Bluetooth Classic, o a través de un servicio GATT privado si está conectado a un dispositivo BLE. Cuando se reciben datos del dispositivo peer desde SPP para Bluetooth Classic o UART Transparente para BLE, dichos datos salen directamente a UART.

El módulo Bluetooth se configura y controla poniéndolo en modo Comando y ejecutando comandos de texto ASCII a través de la UART.

Dispone de un par de modos de funcionamiento que el usuario puede configurar mediante el comando SM:

[Modo por defecto \(SM,0\)](#) - en el que otros dispositivos Bluetooth pueden descubrir y conectarse al módulo. Las conexiones salientes pueden iniciarse en este modo.

Modo de reconexión automática (SM,6) - En este modo, el módulo intenta conectarse al dispositivo remoto que coincida con la dirección remota almacenada. Para configurar la dirección remota, utilice el comando SR.

Todos los cambios de configuración realizados mediante comandos Set permanecen en la memoria no volátil (NVM) y sobreviven al ciclo de encendido. Los cambios de configuración sólo surten efecto tras un reinicio.

Todos los comandos de acción surten efecto inmediatamente, pero no tienen efecto después de la desconexión de la alimentación.

## Ejercicio 6: Servicio de datos serie UART transparente BLE GAP

El RN4678 puede iniciar una conexión BLE en modo Central de Perfil de Acceso Genérico (GAP) con otro dispositivo BLE que admita el servicio "UART transparente".

### Objetivos

- Desarrolla un programa que se conecte a un segundo dispositivo Bluetooth.
- Enviar datos serie al segundo dispositivo
- Imprime los mensajes y datos recibidos en la pantalla LCD del nodo

### Requisitos previos

- Debe utilizar los programas que ha desarrollado anteriormente como punto de partida.
- Comprensión del proceso de investigación y descubrimiento
- Conocer la dirección del dispositivo del módulo al que se va a conectar
- Para los dispositivos que son sólo BLE, utilice el comando "F,5" para preguntar su dirección

### Requisitos de hardware y software

Se necesitan los siguientes elementos de hardware:

- Ambas soluciones Bluetooth son necesarias para este Ejercicio.
- Solución 1, el nodo A estará conectado a

Solución 2, el nodo B conocerá la dirección del nodo A e iniciará la conexión con él

### Información del ejercicio

Comando	Descripción
SG,1	Poner el módulo en modo Bluetooth Low Energy 4.0
SA,2	Modo Secure Simple Pairing (SSP) "Just Works". Este modo funciona sin necesidad de visualizar o introducir ningún secu-
R,1	Reinicie el dispositivo para actuar la nueva configuración.
F,5	Consultar dispositivos BLE en el vecindario
C,<0,1>,<dirección MAC>	Intentar una conexión con el dispositivo remoto, donde el primer parámetro indica el tipo de dirección que puede encontrarse en el resultado de la consulta (0 para dirección pública y 1 para privada dirección)

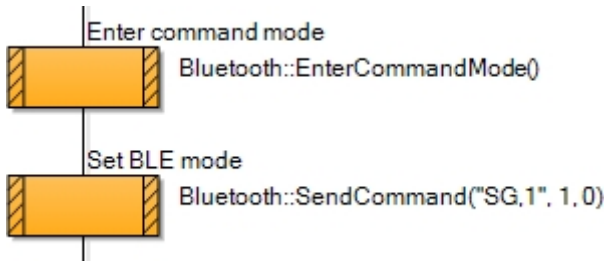
### Resultado del aprendizaje

Los principales resultados de aprendizaje de este ejercicio son

- Configuración del módulo RN4678 para funcionar en modo BLE
- Configuración del módulo RN4678 para funcionar en modo SSP "Just works"
- Conexión a un segundo módulo Bluetooth BLE

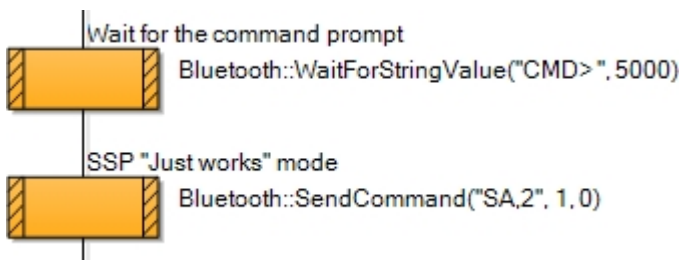
## Modo BLE

Poner el módulo en modo BLE con el comando "SG,1



## Emparejamiento seguro y sencillo

Poner el módulo en modo SSP "Just Works" con el comando "SA,2



## Activar la configuración

Reinicie el módulo para aplicar los ajustes con el comando "R,1".



Para el nodo A, visualización de las respuestas en la pantalla LCD como en los ejercicios anteriores.

Para el nodo B, envía el comando de conexión BLE y el envío de datos serie como en los ejercicios anteriores.

## Teoría: Confianza y seguridad

Las comunicaciones por radio son intrínsecamente vulnerables. Cualquier dispositivo capaz de recibir señales de radio de las mismas bandas de frecuencias puede escuchar la señal. A lo largo de los años, las empresas de radiocomunicaciones han hecho un gran esfuerzo para enviar señales de forma segura. Y también se han dedicado grandes esfuerzos a romper esa seguridad. Puede que Enigma y Bletchley Park sean los dos nombres más famosos de la era de la guerra de las comunicaciones, pero sus descendientes siguen vivos. Lamentablemente, la mayor amenaza para la seguridad no procede de la inteligencia aliada, sino de los más prosaicos pero insidiosos hackers y creadores de virus que plagan Internet. Además, aunque la resistencia francesa no vigile sus señales, tal vez sus rivales comerciales sí lo hagan. La información sobre contactos, acuerdos, ofertas y precios es una munición de valor incalculable para la sala de juntas, y el humilde teléfono, con su arsenal de complementos Bluetooth, está justo en primera línea.

### Seguridad en general

¿Cómo garantiza Bluetooth su seguridad? Existen tres sistemas complementarios.

- Hardware
- Números PIN

Autenticación, cifrado y confianza

A nivel de hardware, Bluetooth emplea técnicas de salto de frecuencia que saltan de una frecuencia a otra. Si se escucha una frecuencia, sólo se obtienen pequeñas partes del mensaje. Para escuchar una conversación completa es necesario conocer las frecuencias a las que hay que saltar, información que no se comparte con los dispositivos que no participan en la comunicación.

Los números PIN se utilizan a menudo en dispositivos como teléfonos móviles, auriculares, etc. El número PIN se suministra con un dispositivo y es necesario introducirlo para que el programa lo compruebe.

Nota: Esto no es lo mismo que la Passkey que se pasa programáticamente, sino un número PIN introducido por el usuario. Los números PIN se utilizan generalmente para informar al dispositivo de que usted es el propietario legítimo para ayudar a evitar que los ladrones utilicen el dispositivo. Los números PIN no deben almacenarse con el dispositivo por razones de seguridad, al igual que no dejaría su número PIN con sus tarjetas bancarias.

A nivel de programa y de dispositivo, las comunicaciones pueden ser autenticadas y el dispositivo puede ser registrado como dispositivo de confianza.



# Modos de seguridad

El módulo RN4678 soporta encriptación y autenticación con modos de seguridad de 1 a 4. La definición del modo de seguridad es la siguiente:

- Modo de seguridad 1: Passkey Confirm
- Modo de seguridad 2: Sólo funciona
- Modo de seguridad 3: Introducción de claves de acceso
- Modo de seguridad 4: Código PIN heredado

Para Bluetooth Classic, se admiten todos los modos de seguridad. Para BLE, solo se admiten los modos de seguridad 1 a 3.

Para los modos de seguridad 1 y 3, por defecto, se genera un pin de seguridad aleatorio de 6 dígitos que se muestra en un extremo de la conexión y el otro extremo debe introducir el pin de seguridad. Opcionalmente, si ambos extremos de la conexión son RN4678, es posible fijar el pin de seguridad de 6 dígitos para BLE suministrando el pin de 6 dígitos al comando SP en lugar del pin de 4 dígitos para el modo de código PIN heredado.

## Autenticación

El comando Set Authentication del módulo RN4678 establece el método de autenticación cuando un dispositivo remoto intenta conectarse. Una vez que un dispositivo remoto intercambia códigos pin con el dispositivo RN4678, se almacena una clave de enlace para su uso futuro. El dispositivo almacena de forma automática y permanente hasta cuatro dispositivos pares en la memoria flash utilizando el método FIFO (primero en entrar, primero en salir).

Se utiliza un código pin de seguridad de 4 dígitos para el emparejamiento de código pin heredado, mientras que el pin de 6 dígitos se utiliza para la autenticación SSP en BLE si se desea un pin fijo. El código pin fijo de 6 dígitos en BLE no es compatible con muchas de las implementaciones de Bluetooth Low Energy.

## Confíe en

Cuando dos dispositivos Bluetooth han instigado el emparejamiento y establecido comunicaciones entre ellos, se dice que forman un par de confianza. Una vez que se ha establecido un par de confianza, pueden comunicarse entre sí sin necesidad de descubrimiento o autenticación. Como es obvio que esto acelerará los procesos de conexión y comunicación, hay una serie de comandos y opciones disponibles para controlar la grabación de los dispositivos en los que confías.

Los dispositivos de confianza se almacenan en una lista. La lista de dispositivos de confianza se almacena en la memoria no volátil, lo que permite conservarla durante el apagado. Por otro lado, la caché es temporal y sólo incluye el último dispositivo que se emparejó y obtuvo la confianza.

## Ejercicio 7: Confianza y seguridad

### Introducción

Una buena comunicación es a la vez segura y eficaz. La seguridad es necesaria para bloquear las comunicaciones no autorizadas. Sin embargo, la seguridad no debe ser demasiado intrusiva, y los dispositivos de confianza deben poder comunicarse con facilidad.

### Objetivos

Crea un programa que realice una conexión autenticada con otro dispositivo Bluetooth.

Dale al dispositivo tres opciones controladas por interruptores en el puerto A:

- Lista de dispositivos actualmente en la lista de dispositivos de confianza
- Añadir el dispositivo actual a la lista de dispositivos de confianza
- Elimina el dispositivo actual de la lista de dispositivos de confianza.
- Utilice el programa para demostrar el uso de la lista de dispositivos de confianza.

### Requisitos previos

- Comprensión del proceso de conexión por parejas detallado en el Ejercicio 4.
- Comprensión de las respuestas detalladas en el Ejercicio 5.

### Requisitos de hardware y software

Se necesitan los siguientes elementos de hardware:

- Ambas soluciones Bluetooth son necesarias para este Ejercicio.
- El nodo A se utilizará como dispositivo para añadir o eliminar de la lista de dispositivos de confianza.
- El Nodo B se utilizará para alcanzar los objetivos.

### Información sobre el ejercicio

Comando	Descripción
SP,<código de 6 dígitos>	Establecer el código PIN de seguridad
SA,1	Modo de emparejamiento simple y seguro (SSP)
Y	Mostrar direcciones de dispositivos en la lista emparejada con índice desde 1 a 8
C,<1-8>,<MAC ad-vestido>	Intentar una conexión con el dispositivo remoto
K,1	Desconecte
U,< 1-8>	Eliminar un dispositivo de la lista de dispositivos emparejados
U,Z	Eliminar todos los dispositivos de la lista de dispositivos emparejados

Para más detalles sobre los comandos, consulte la Referencia de comandos de la Guía del usuario del módulo.

### Resultado del aprendizaje

Los principales resultados de aprendizaje de este ejercicio son

Principios de seguridad, entre otros:

- Autenticación
- Dispositivos de confianza

Usando la lista de Dispositivos de Confianza incluyendo:

- Añadir dispositivos
- Retirar dispositivos
- Dispositivos de listado

# Aplicación práctica: Confianza y seguridad

## Objetivos generales

El programa del nodo A actuará como el dispositivo remoto pasivo con el que se emparejará y conectará. Por lo tanto, este nodo se configura en modo esclavo BLE con los comandos "SG,1" y "SM,0" y el código pin de seguridad se configura con el comando "SP,123456". A continuación, el dispositivo requiere un reinicio con el comando "R,1" para configurar estos ajustes.

Para ayudar a identificar este nodo Bluetooth utilizamos el comando "GB" para obtener y luego mostrar la dirección del dispositivo.

El nodo B es el programa activo y se configura en modo maestro BLE con los comandos "SG,1" y "SM,1". El emparejamiento simple seguro (SSP) se configura con el comando "SA,1" y el código pin con el comando "SP,123456".

A continuación, el programa principal debe entrar en un bucle continuo que introduzca el estado de los interruptores y responda a cualquier entrada, enviando el comando adecuado según sea necesario.

Para este ejercicio usa el Interruptor 0 como 'Listar dispositivos de confianza', el Interruptor 1 como 'Añadir dispositivo y conectar', el Interruptor 2 como 'Desconectar' y el Interruptor 3 como 'Eliminar dispositivo'.

Los comandos que hay que enviar son:

Interruptor	Comando	Descripción
Interruptor 0	Y	Obtener y mostrar la lista de dispositivos de confianza
Interruptor 1	C,<1-8>,<dirección MAC>	Emparejar y conectar con un dispositivo, añadir a la lista
Interruptor 2	K,1	Desconectar del dispositivo remoto
Interruptor 3	N,<1-8>	Eliminar la dirección del dispositivo remoto de la lista

Una forma sencilla de determinar si ha funcionado Añadir y quitar dispositivo de la lista de dispositivos de confianza consiste en listar los dispositivos y buscar la dirección del dispositivo.

La pantalla LCD se puede utilizar para mostrar no sólo la Lista de Dispositivos de Confianza, sino también para mostrar mensajes de respuesta para las otras opciones.

## Características adicionales

Como un extra opcional el programa base puede ser extendido para añadir dispositivos adicionales y también para borrar toda la lista de Dispositivos de Confianza. Encontrará más detalles sobre estos comandos en la sección

"Guía del usuario del módulo de modo dual Bluetooth® 4.0 RN4678".

# Principios de diseño del proyecto

Una vez que los alumnos se hayan familiarizado con Bluetooth y hayan realizado los ejercicios de este curso, la siguiente etapa consistirá en utilizar esos conocimientos para diseñar y ejecutar un proyecto.

A continuación se exponen consideraciones sobre el diseño, pero no se dan códigos ni instrucciones. Pueden utilizarse como punto de partida para crear los objetivos y especificaciones del proyecto.

## Proyecto - Mando a distancia sencillo

Un uso sencillo de la transferencia de datos podría ser el control inalámbrico de un dispositivo. Esto se puede demostrar mediante el uso de una placa de interruptores conectada a un procesador y un módulo Bluetooth configurado como maestro para transmitir los datos del interruptor a un segundo procesador con un módulo Bluetooth configurado como esclavo que recibe los datos y los repite en una placa LED conectada.

### Ampliar el proyecto

Considera cómo reducir al mínimo la cantidad de transferencias de datos.

## Proyecto - Datalogger médico

Una tarea útil para Bluetooth sería el registro de datos médicos. Los pacientes necesitan monitorización, pero los cables y enchufes atan al paciente a la cama y suponen un trabajo extra para el personal. Si un paciente necesita hacerse pruebas o incluso ir al baño, hay que desenchufar y mover el equipo, o éste debe ser móvil. ¿Y qué hay más móvil que el Bluetooth? Para los pacientes, la posibilidad de moverse les da más libertad e independencia. Para el personal, el sistema ofrece registro de datos las 24 horas del día y facilita el movimiento del paciente. También se pueden utilizar alertas de datos para avisar al personal de los problemas que surjan.

Estos registradores de datos también ayudarían en las pruebas médicas, como en el registro de datos relacionados con el deporte, donde los sensores tradicionales con cable no harían más que estorbar.

Lo primero que debe tener en cuenta para el registrador de datos remoto es qué datos desea registrar.

Con un simple registrador de datos se podría controlar la temperatura. También se pueden añadir otros sensores, como monitores de frecuencia cardíaca y respiración.

A continuación, hay que tener en cuenta la frecuencia con la que deben actualizarse los datos. Para algo como la temperatura, puede ser cada 5 o 10 minutos. Para algo más crítico, como la monitorización de la frecuencia cardíaca, puede necesitar lecturas cada pocos segundos. Después de decidir la frecuencia de envío de los datos, hay que decidir la estrategia de envío.

¿Mantener un contacto continuo? ¿Contactar y enviar los datos? ¿Recoger datos durante un tiempo y luego enviarlos por lotes?

A continuación, ¿hay que tener en cuenta algún caso especial? ¿Es importante un aumento repentino de la frecuencia cardíaca? ¿Y la temperatura? ¿Hay que avisarnos con luces intermitentes y timbres de alarma? ¿Deben sonar nuestros pitidos para que sepamos que hay un problema? ¿O nos basta con una simple alerta LED y un mensaje en una pantalla LCD? ¿Necesitamos saberlo ahora o dentro de 5 minutos, en la próxima emisión de datos programada?

### Ampliar el proyecto

Existen dos problemas obvios con los monitores médicos en comparación con los monitores fijos que restringen el movimiento.

- ¿Qué ocurre cuando el paciente sale del radio de acción de la red? ¿Dónde está el paciente cuando salta la alarma?

¿Cómo abordar estos problemas?

Algunas de estas cuestiones son demasiado importantes para abordarlas en un proyecto, pero mostrarían cómo se han tenido en cuenta si las redactara el estudiante en una sección de "Características ampliadas/adicionales" del informe del proyecto.

