

Uso de módulo HM-10 BLE como Beacon para bastón guía inteligente

Informe de Avance

Grupo N° 19

Integrantes

Delgado, David – 00139/8

Silva, Agustín – 00063/5

Fecha: 10 de Noviembre de 2016

Índice

1. Introducción	2
2. Propuesta original	2
3. Materiales Utilizados	2
4. Esquema gráfico del proyecto completo propuesto	3
5. Identificación de partes	3
a. E/S del controlador con el exterior, excepto PC	3
b. Comunicaciones con la PC	3
c. Sistema web	3
d. Infraestructura de software propuesta para la PC	4
6. Solución Implementada	4
a. Conexión	4
b. Subir el código Arduino a la placa	5
c. App para celular	6
i) Introducción	6
ii) Diseño	7
iii) Comentarios a futuro	8
7. Problemas encontrados	8
Limitaciones del módulo	8

1. Introducción

Nuestro proyecto es parte de una propuesta de mayor tamaño que engloba un bastón guía inteligente para invidentes, el cual facilitaría su vida diaria, no solo permitiéndole movilizarse con mayor facilidad sino también al proveerle información de su entorno que normalmente no podría conseguir sin tener que pedirle asistencia a otra persona.

2. Propuesta original

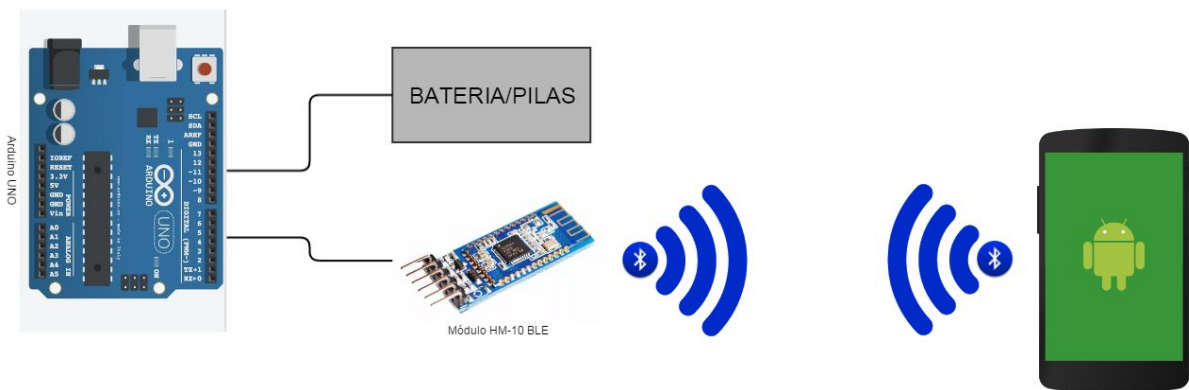
El objetivo de este proyecto es implementar un Beacon (y en un futuro una red de los mismos) que se comunica con un bastón guía inteligente para individuos que sufren de ceguera, enviando información acerca del entorno, como por ejemplo, el estado de un semáforo o los nombres de las calles donde se encuentra actualmente.

3. Materiales Utilizados

Nombre	Cantidad
Arduino UNO + cable USB	1
Módulo HM-10 BLE	1
Pilas/Bateria para alimentación	Por el momento no es necesario
Porta pilas AA	Por el momento no es necesario
Contenedor hermético	Por el momento no es necesario
Cables macho-hembra	4
Smartphone con Android 4.4.2 o superior	1

PC	1
----	---

4. Esquema gráfico del proyecto completo propuesto



5. Identificación de partes

a. E/S del controlador con el exterior, excepto PC

El módulo HM-10 BLE se encontrará conectado a la placa Arduino UNO, el cual se encontrará funcionando debido a la energía proporcionada por las baterías AA.

b. Comunicaciones con la PC

La placa Arduino UNO se conectará a la PC solo durante la etapa de desarrollo del proyecto ya que posteriormente ésta contendrá el programa almacenado y recibirá alimentación de las baterías.

c. Sistema web

Aunque el sistema no incorpora conexión a un servidor web de por si, los beacons tienen la posibilidad de enviar una página web, de la cual el bastón inteligente podría levantar información.

d. Infraestructura de software propuesta para la PC

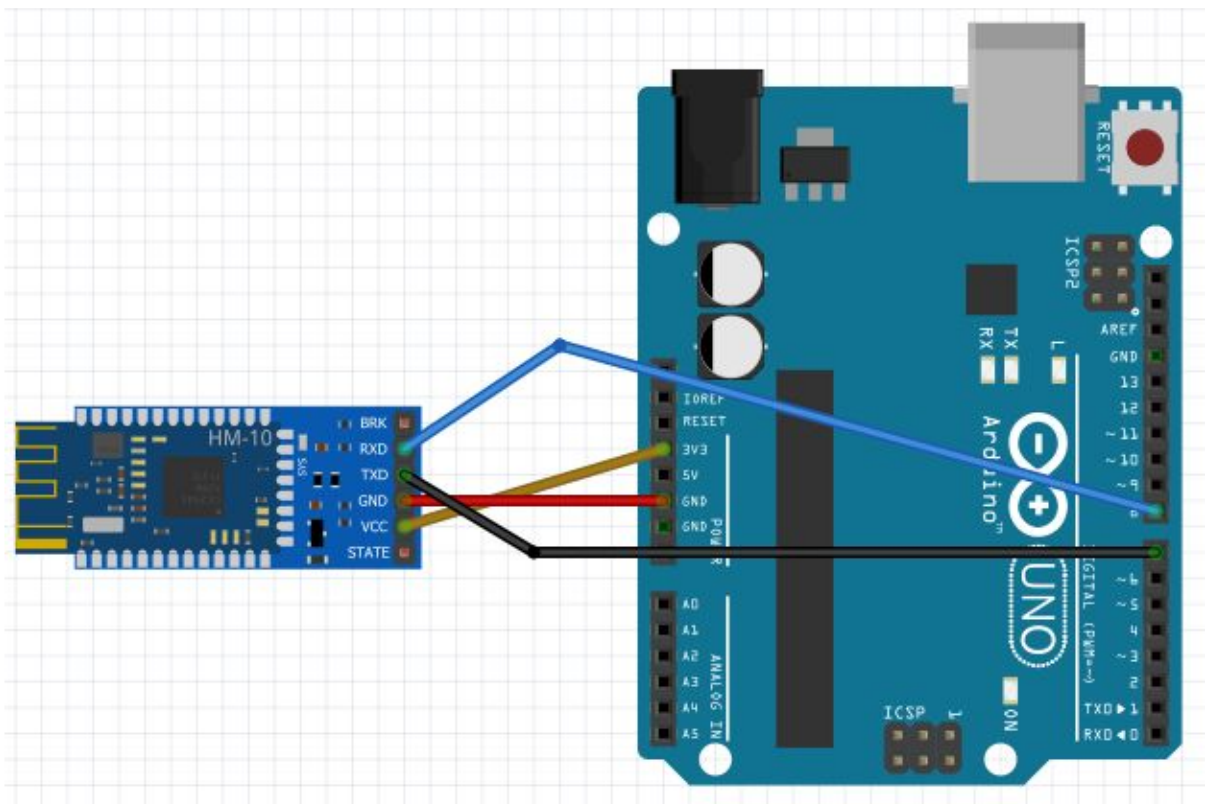
Como se mencionó en el ítem anterior, nuestra propuesta no necesita un servidor web conectado.

6.Solución Implementada

a. Conexión

Conectar los componentes de la siguiente manera:

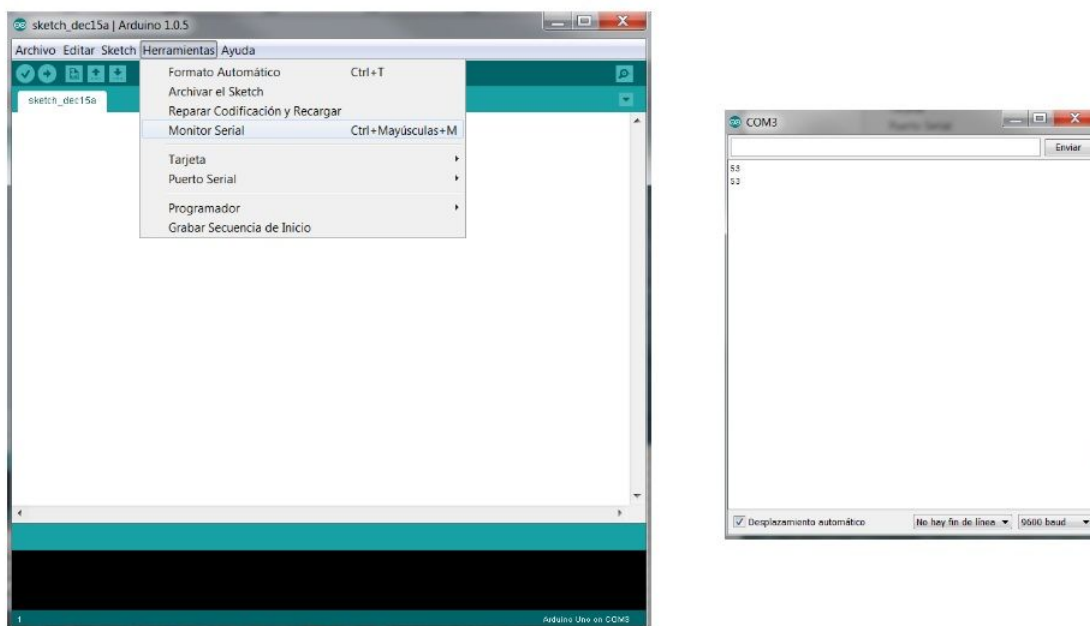
1. Conectar la alimentación 3.3V del Arduino al VCC del HM-10
2. Conectar la tierra del Arduino al GND del HM-10
3. Conectar el D8 del Arduino al RX del HM-10
3. Conectar el D7 del Arduino al TX del HM-10



b. Subir el código Arduino a la placa

Para poder subir el código al controlador, primero se debe descargar el IDE¹ (entorno de desarrollo integrado) de Arduino según el sistema operativo utilizado. Una vez descargado e instalado, conectar la placa Arduino UNO a la PC mediante un cable USB, iniciar el IDE y abrir el sketch de Arduino llamado “BeaconScanner.ino”. Luego, compilar el sketch para corroborar de que todo esté bien (“Programa > Verificar/Compilar”) y subirlo al controlador (“Programa > Subir”).

Una vez subido, hay que configurar el módulo HM-10 de forma tal que funcione como un Beacon. Para ello, abrimos el “Monitor Serial” ubicado en “herramientas > monitor serial” o presionar “ctrl+shift+m”.



En la ventana que emerge, ejecutar uno por uno los siguientes comandos:

- | | |
|---------------------------|--|
| 1. AT+RENEW | Restauración de fábrica |
| 2. AT+RESET | Resetea y guarda cambios realizados |
| 3. AT | Devuelve OK si reconoce al módulo |
| 4. AT+MARJ0X002F | Calles entre 32 a 72 en hexadecimal |
| 5. AT+MINO0X0001 | Calles entre 1 a 31 en hexadecimal |
| 6. AT+ADVI5 | Establece el intervalo de refresco a 5 (546.25 ms) |
| 7. AT+NAMESEMAFORO | Establece el nombre del módulo HM-10 |
| 8. AT+ADTY3 | Lo configura como no-conectable (ahorra energía) |

¹ <https://www.arduino.cc/en/Main/Software>

9. AT+IBEA1	Habilita el modo Beacon
10. AT+DELO2	Modo Beacon solo transmisión (ahorra energía)
11. AT+RESET	Resetea y guarda los cambios realizados

c. App para celular

i) Introducción

La primera versión de la aplicación para celular por ahora solo soporta un beacon y su función es imprimir en pantalla los datos que lee del mismo, estos están contenidos en el “Scan Response Package Data”, que es el paquete que envía el dispositivo cuando es descubierto por la aplicación, este contiene principalmente:

- Nombre del módulo: Nombre que se le asignó al módulo.
- Tx: Byte que define la potencia con la que le envía el paquete.
- Mayor y Menor: Cada uno está compuesto por 2 bytes, pueden ser asignados por el programador al configurar el módulo.
- Address: Dirección MAC del módulo.
- FLAG: Este módulo solo permite editar un bit de estado, el cual es cambiado periódicamente por el arduino al cual está conectado el mismo.

La aplicación al leer el paquete, además lee la intensidad de la señal al recibirlo y junto con el valor de Tx del paquete puede calcular la distancia al módulo.

Además, lee el FLAG y lo utiliza para determinar el estado del semáforo, si el FLAG está en 0, el semáforo está en VERDE, caso contrario, el semáforo está en ROJO. En esta primera instancia el Arduino intercambia los valores del FLAG periódicamente para demostrar el funcionamiento del mismo. Cuando se vaya a implementar en la calle, se deberá asegurar que el Arduino esté sincronizado con el semáforo real, o de ser posible, que lea el estado del mismo y configure el FLAG.

ii) Diseño

Estado inicial de la aplicación:

Beacon Scanner

Nombre:

Potencia:

Mayor:

Menor:

Semáforo: Estado

Direccion:

ESCANEAR

DETENER

Escanear

Estado:

Estado luego de descubrir el módulo:

Beacon Scanner

Nombre: semaforo

Potencia: -53 | Distancia: 0.34 m.

Mayor : 47

Menor: 1

Semáforo: **ROJO**

Direccion: 78:A5:04:3E:0D:53

ESCANEAR

DETENER

Detener

Estado: Escaneando

Como se aprecia en las imágenes y como se explicó anteriormente, la aplicación por ahora se limita a imprimir en pantalla la información del módulo. En este caso se utiliza el Mayor y el Menor para almacenar las calles y el FLAG el estado del semáforo. Además muestra la distancia aproximada del celular al módulo.

La aplicación posee 3 botones, el primero escanea solo por 5 segundos y luego se detiene, el segundo botón sirve para forzar la detención del escaneo, y el último botón, que posee 2 estados, permite escanear constantemente hasta que se vuelva a presionar.

iii) Comentarios a futuro

En una versión final solo se utilizará el último botón, activado por voz o desde el bastón guía. Además locales comerciales podrían incorporar sus propios beacons, que posean su dirección almacenada en los registros Mayor y Menor, y el nombre de su local. La aplicación podría comunicarse con un servidor donde se almacene y compare esta información y al acercarse al local podría levantar desde el servidor la página web del local o una página específica que contenga las ofertas del día por ejemplo.

7. Problemas encontrados

a. Limitaciones del módulo

La potencia recibida del módulo HM-10 fluctúa notablemente, lo cual conlleva a un error en el cálculo de la distancia del módulo al celular y este es un factor importante ya que la aplicación está destinada para una persona no vidente, y una diferencia significativa en este valor calculado puede llegar a ocasionar potenciales accidentes.

Además cuando el módulo esta conectado con el semáforo debe siempre estar recibiendo comandos desde el arduino, por lo cual no puede entrar en modo “auto-sleep” y por lo tanto consume una cantidad de energia superior a la que estos dispositivos BLE (Bluetooth Low Energy) deberían funcionar.