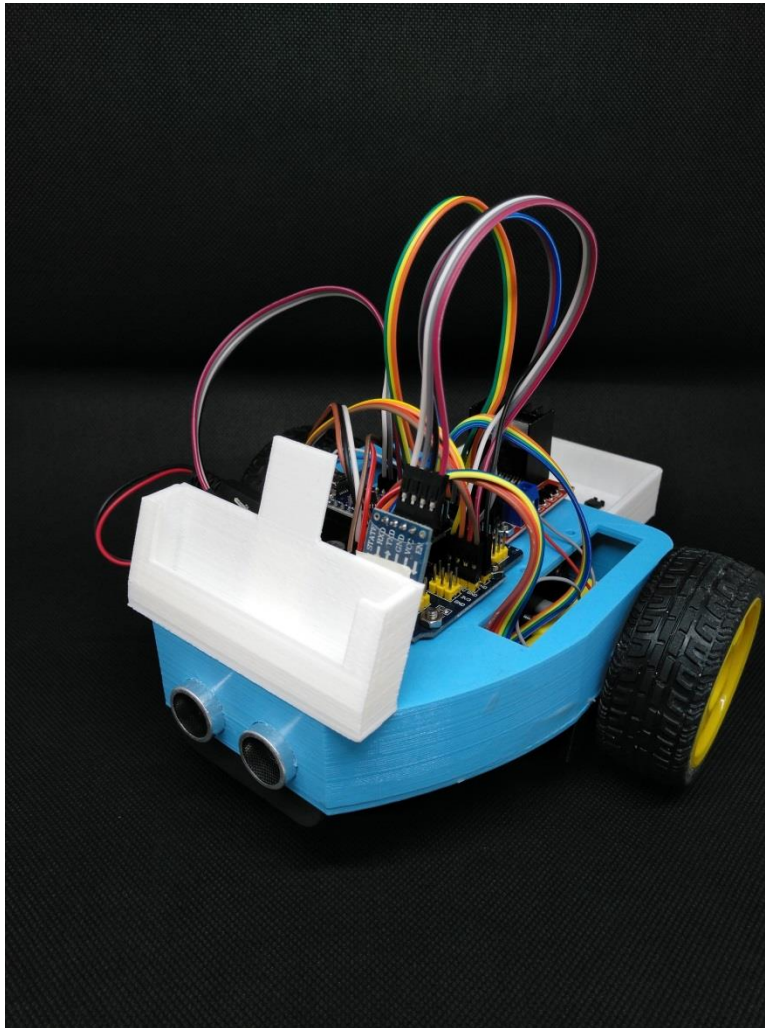


RETO ASCENTIC 2019

ASCENTICBOT



MANUAL DE MONTAJE Y PROGRAMACIÓN ASCENTICBOT

Práctica 7: BLUETOOTH Y APP

Parte I: Bluetooth

Para desarrollar el **RETO ASCENTIC**, es necesario implementar una comunicación bidireccional Bluetooth entre el Arduino y la APP del teléfono móvil. Para ello se proporciona con el kit AscenticBot un módulo Bluetooth HC-06.



Todos los mensajes son de texto. No existe ningún mecanismo que se haya programado en la APP que asegure la recepción correcta de un mensaje enviado hacia o desde Arduino.

En este apartado se explica la programación y el conexionado en la parte del Arduino así como el protocolo de comunicaciones y la secuencia de utilización de la APP.

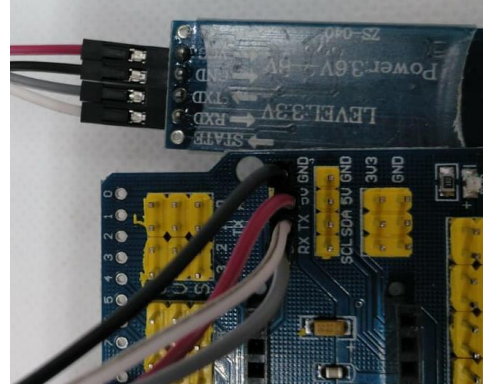
En lo que se refiere a la programación de la APP se puede consultar la entrega correspondiente en la página web www.cantabrobots.es desde donde podrás descargarte tanto el archivo “.apk” (instalable) como el archivo “.aia” (proyecto de app inventor).

Práctica 7: BLUETOOTH Y APP

Parte II: Conexión en Arduino del modulo Bluetooth

El módulo de comunicaciones Bluetooth tiene 4 conexiones:

- Vcc** : Alimentación 5V (+).
- Gnd**: Alimentación 0V (-).
- Tx**: Transmisión de datos.
- Rx**: Recepción de datos.



Para disponer de una conexión Bluetooth operativa es necesario realizar las conexiones como sigue:

CABLE (Colores)	ARDUINO	MODULO BLUETOOTH (HC06)
Rojo	Vcc – 5V	Vcc
Negro	GND	Gnd
Gris	Tx – Pin digital 0	Rx
Blanco	Rx – Pin digital 1	Tx

Práctica 7: BLUETOOTH Y APP

Parte III: Comandos a intercambiar.

El protocolo de comunicación entre el Arduino y la APP incorpora los siguientes mensajes:

EMISOR	RECEPTOR	COMANDO	ACCION
App	Arduino	e	Iniciar reto.
Arduino	App	o	Voz-obstáculo, disculpe las molestias.
Arduino	App	h	Voz-Deposite su maleta.
Arduino	App	f	Voz- fin de reto.
Arduino	App	a	Voz-Cuevas Altamira.
Arduino	App	p	Voz-palacio de la Magdalena.
Arduino	App	s	Voz-Cuevas del Soplao.
Arduino	App	i	Voz-Picos de Europa.
Arduino	App	r	Voz-Santillana del Mar.
Arduino	App	c	Voz-Parque Cabárceno.

Práctica 7: BLUETOOTH Y APP

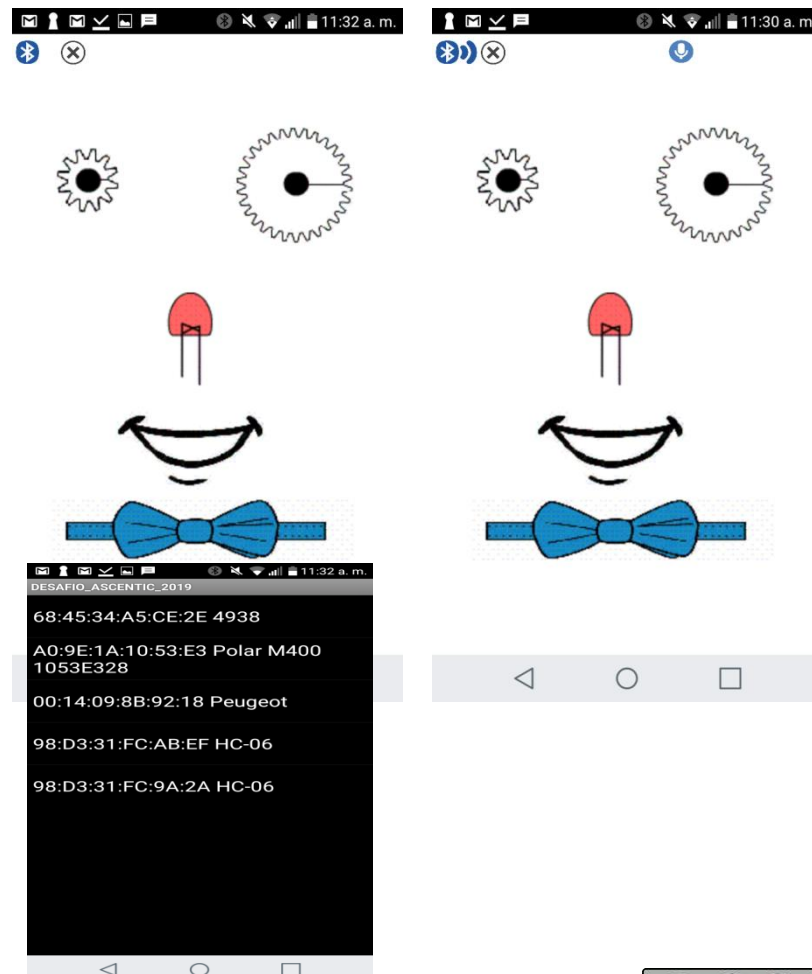
Parte IV: Secuencia de comunicaciones.

El modo en el que están diseñadas las pruebas así como el modo en el que está desarrollada la APP obliga a una secuencia concreta durante el intercambio de mensajes.

Paso 1: Habilitar conexión Bluetooth. Para esto será necesario previamente tener activado el Bluetooth en el móvil y vinculado el dispositivo HC-06.

Posteriormente se debe iniciar la APP, pulsar el botón de Bluetooth y seleccionar el módulo HC-06.

Una vez realizada la conexión con éxito se indica mediante el icono Bluetooth y se habilita un nuevo botón en la parte superior (micrófono) que permite iniciar el paso 2 de reconocimiento de voz.

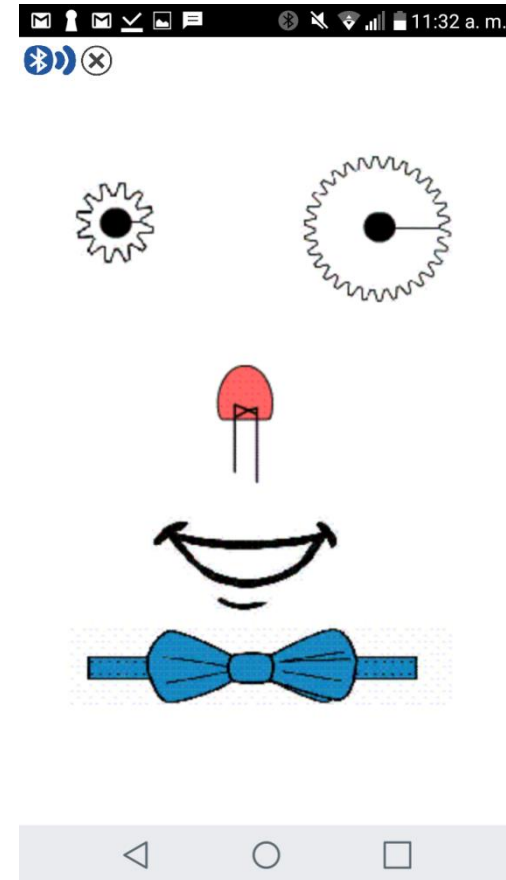


Práctica 7: BLUETOOTH Y APP

Parte IV: Secuencia de comunicaciones. (Continuación)

Paso 2: Envío desde la APP del mensaje de comienzo de reto “e”. Para ello es necesario utilizar el reconocimiento de voz de la APP diciendo en voz alta la palabra “Empezar”. Hasta que este mensaje no se envíe a través de la APP no se procesará desde esta ningún mensaje que pueda haber sido enviado desde Arduino.

Paso3: Envío desde Arduino de comandos a la APP de acuerdo al protocolo establecido en el apartado II.



Práctica 7: BLUETOOTH Y APP

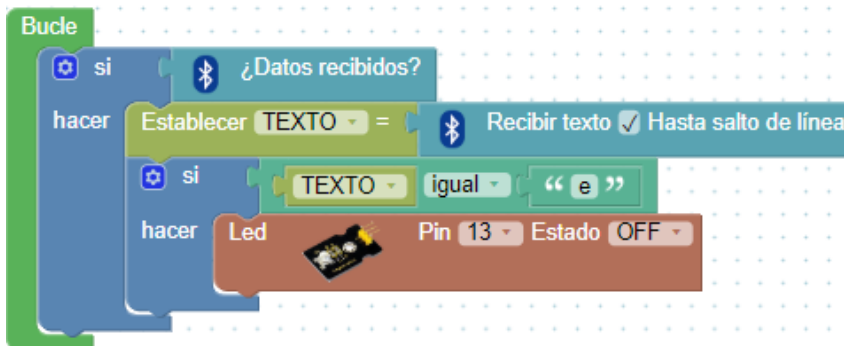
Parte V: Inicio de la comunicación Bluetooth.

Inicio de la comunicación Bluetooth.



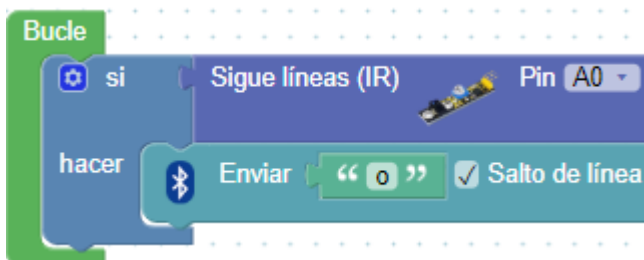
Parte V: Recepción de mensajes.

Recepción de mensajes.



Parte VI: Envío de mensajes.

Envío de mensajes.



Parte VI: Programa de test de protocolo de comunicación.

A continuación se muestra un programa que sirve para comprobar el protocolo de comunicaciones entre el Arduino y la APP.

En este programa, se inicia la comunicación en los pines 0 y 1 (Rx/Tx) y se pone el led RGB inicialmente a rojo. (Bloque inicializar)

Posteriormente en el bucle, se está esperando continuamente el mensaje “e”. Al recibirlo se cambia el led a verde. En el momento en que se activa un sensor siguelíneas colocado en el pin A0, se inicia el envío de un carrusel que contiene todos los mensajes del protocolo, con un tiempo de 5 s. entre mensajes para que a la APP le de tiempo a reproducir la voz completa.

The image shows a screenshot of an Arduino IDE sketch. The sketch is written in a block-based language (likely Scratch or Blockly) and is organized into three main sections: 'Inicializar', 'Bucle', and 'Enviar'. The 'Inicializar' section sets up the serial port (Rx 0, Tx 1) and initializes three LEDs (Pin 13, 12, 11) to OFF. The 'Bucle' section is a loop that checks for received data. If it receives the character 'e', it turns the LED on Pin 13 ON. If it receives 'h', it turns the LED on Pin 12 ON. If it receives 'f', it turns the LED on Pin 11 ON. If it receives 'E', it sends a series of characters 'h', 'p', 'E', 'i', 'r', 'r', 'e' to the serial port, each with a 5000ms delay. The 'Enviar' section is a separate block that sends the character 'e' to the serial port with a 5000ms delay.