Proyecto TDRCI: Comunicaciones Bluetooth

Aplicación Práctica

Miguel Cantón Cortés

Jesús Linares Bolaños

Contenidos

1. Objetivo

- El fin de este proyecto es aprender sobre la tecnología bluetooth mediante una aplicación práctica.
- Para ello, hemos desarrollado un sistema basado en la comunicación entre un microcontrolador, un móvil y un PC.

2. Base teórica: Bluetooth
Introducción

- Especificación industrial para Redes Inalámbricas de Área Personal (WPANs).
- Permite la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia.
- Opera en la banda ISM (reservada internacionalmente para uso no comercial) de los 2,4 GHz.
- El protocolo Bluetooth está diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja y basados en transceptores de bajo costo.

2. Base teórica: Bluetooth
Clasificación

Clasificación según potencia:

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~10 metros
Clase 3	1 mW	0 dBm	~1 metro

Clasificación según ancho de banda:

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s
Versión 4.0	24 Mbit/s

2. Base teórica: Bluetooth. Especificaciones

- **Bluetooth 1.x:**
  - La última versión, v1.2, es compatible con USB 1.1 e introdujo algunas mejoras como control de flujo, una conexión inicial más rápida y mejoras en aplicaciones de audio.
- **Bluetooth 2.x + EDR:**
  - Es compatible con la especifiación v1.2. Añade una velocidad de datos mejorada (EDR) y la posibilidad de ahorrar energía funcionando a menor velocidad.
- **Bluetooth 3.0 + HS:**
  - Permite el tráfico de datos a alta velocidad usando un enlace 802.11.
- **Bluetooth 4.0:**
  - A la alta velocidad conseguida por la especificación 3.0 + HS, se añaden mejoras de ahorro energético.

2. Base teórica: Bluetooth. Arquitectura Hardware

- Un dispositivo Bluetooth está compuesto por:
  - **Capa física:** dispositivo de radio, encargado de modular y transmitir la señal.
  - **Capa de enlace:** un controlador digital, compuesto por una CPU, por un procesador de señales digitales y de las interfaces con el dispositivo anfitrión.

2. Base teórica: Bluetooth.
Usos

- La principal ventaja de Bluetooth es su simplicidad a la hora de establecer enlaces. Esto unido a su reducido ancho de banda en las especificaciones tradicionales y cobertura baja pero buen rendimiento energético, lo hace idóneo para pequeños dispositivos inalámbricos como ratones, impresoras o PDAs.

3. Aplicación
Introducción

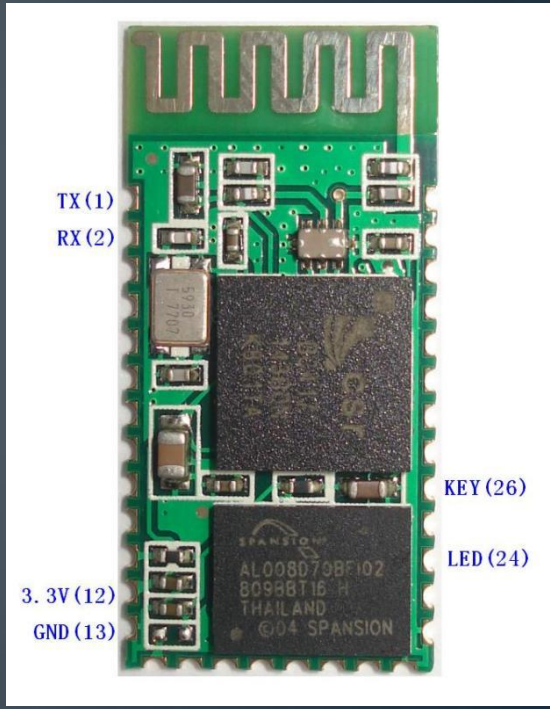
- Nuestro objetivo es comunicar un microcontrolador, un PC y un móvil mediante Bluetooth. Para ello, hemos realizado una aplicación que simula un robot detector de minas.
- El robot recorre un terreno de manera semiautomática. Los límites del terreno y las minas son detectados de manera automática mediante sensores. Los giros son controlados mediante un móvil.
- Una vez analizado el terreno en busca de minas, los resultados son enviados a un PC.
- El software genera un mapa del terreno con la ubicación de las minas a partir de los datos recibidos del robot.

3. Aplicación Hardware

1. Módulo Bluetooth
2. Arduino
3. Barra Sensora
4. Vehículo

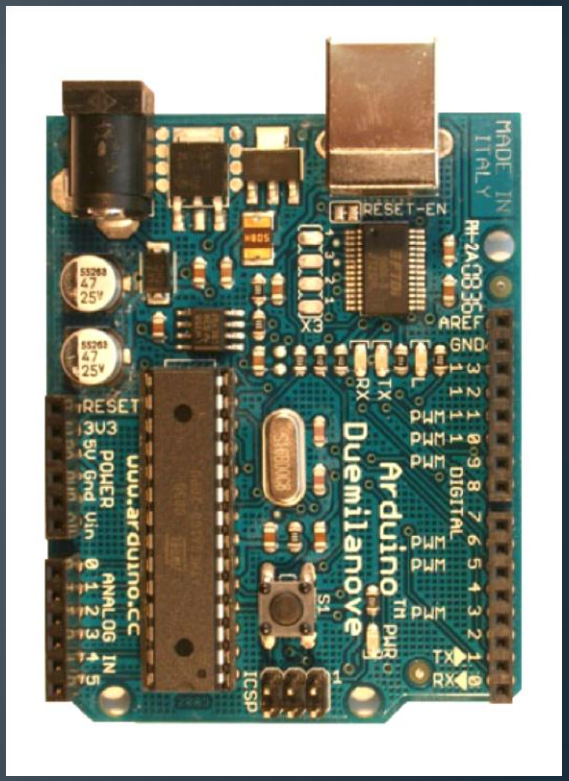
3. Aplicación
Hardware: Módulo Bluetooth

- Módulo Bluetooth RS232 TTL
- Permite comunicación serie usando tecnología TTL (*transistor-transistor logic*). Esto lo hace idóneo para trabajar con un microcontrolador.
- Chipset: CSR
- Bluetooth V2.0
- Voltaje: 3.3V
- Baud Rate: 1200, 2400, 4800, **9600**, 19200, 38400, 57600, y 115200.



3. Aplicación
Hardware: Arduino

- Microcontrolador: ATmega368 (16 MHz)
- Voltaje de funcionamiento: 5V
- Voltaje de entrada: 7-12V
- Pines E/S digitales: 14 (6 proporcionan salida PWM)
- Pines de entrada analógica: 6
- Memoria Flash: 2 KB para el bootloader y 30KB para el programa.
- SRAM: 2 KB
- EEPROM: 1 KB



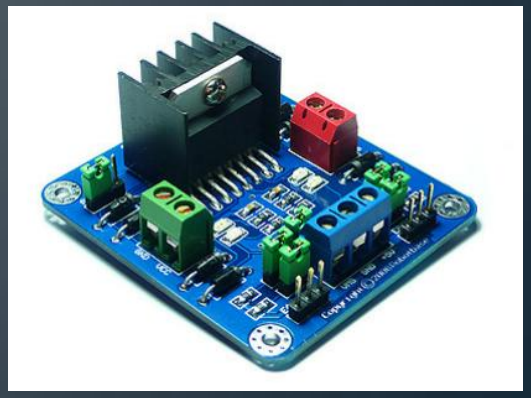
3. Aplicación
Hardware: Barra Sensora

- Sensor de reflectancia: QTR-8RC
- Voltaje: 3.3-5.0 V
- E/S digitales: 8
- Distancia óptima: 3 mm
- Distancia máxima recomendada: 9.5 mm

3. Aplicación
Hardware: Vehículo

- Chasis
  - Alimentación motores: 7.5 V (5 pilas AA)
  - Velocidad: 90cm/s
  - Dimensiones: 200x170x105mm
  - Peso: 660 gramos (Sin pilas)

- Controlador de Motores (L298)
  - Convierte las señales del microcontrolador en niveles de voltaje aplicados a los motores.

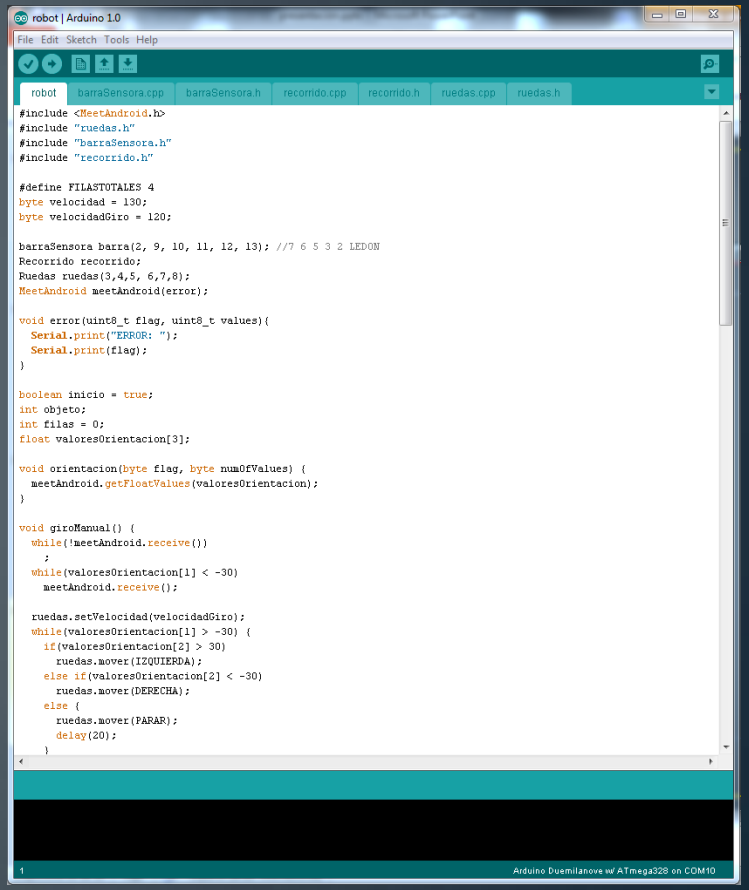


3. Aplicación Software

1. Arduino IDE
2. Android: Amarino y MeetAndroid
3. Processing

3. Aplicación
Software: Arduino IDE

- El lenguaje Arduino está basado en C/C++ y soporta todas las construcciones de C estándar y algunas funcionalidades de C++.
- Sketch: nombre que usa Arduino para un programa.
- Funciones especiales:
  - setup(): es llamado una vez, cuando comienza el sketch. Para realizar tareas de configuración.
  - loop(): Bucle de ejecución.



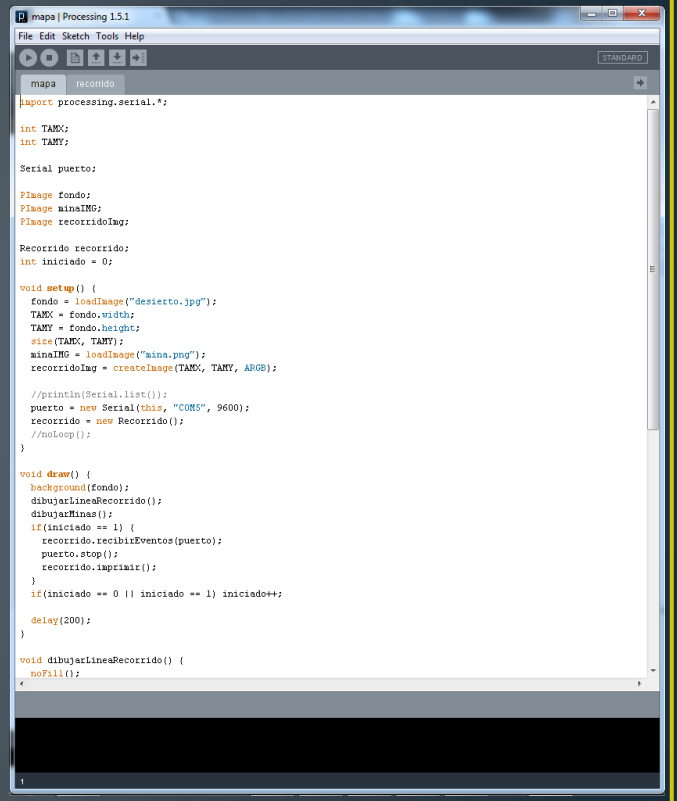
3. Aplicación
Software: Amarino y MeetAndroid

- Amarino
  - Es una aplicación para Android que nos permite obtener información de los sensores del móvil y enviarla en forma de eventos por Bluetooth.

- MeetAndroid
  - Es una librería para Arduino que permite asociar los eventos generados en el móvil a nuestras propias funciones.

3. Aplicación
Software: Processing

- Es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java.
- Hereda toda la funcionalidad de Java.
- Aporta librerías que facilitan el desarrollo de aplicaciones.
- Funciones especiales:
  - setup(): es llamado una vez, cuando comienza el sketch. Para realizar tareas de configuración.
  - draw(): bucle de ejecución.



3. Aplicación
Desarrollo

1. Vehículo
2. Barra sensora
3. Bluetooth
4. Construcción del terreno
5. Programa Arduino
6. Programa PC

3. Aplicación
Desarrollo: Vehículo

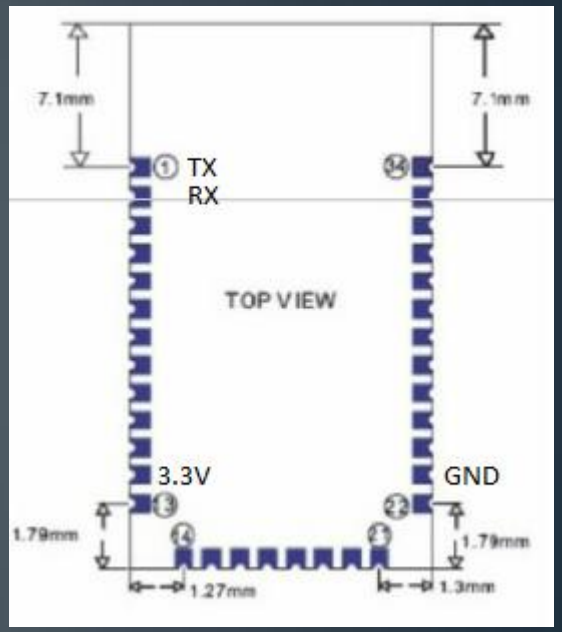
- Chasis
  - Montamos el chasis y el circuito de corriente de los motores.
  - Añadimos cintas a las ruedas para obtener mayor adherencia a la superficie.
- Controlador de motores
  - Conectamos el controlador de los motores al Arduino.
  - Creamos una librería con funciones básicas para mover el vehículo.

3. Aplicación
Desarrollo: Barra sensora

- Conectamos 5 de los 8 sensores al arduino.
- Creamos una librería para obtener valores de los sensores.
- Algoritmo de medición:
  - Encendemos los LEDs infrarrojos.
  - Ponemos en 1 lógico el sensor a medir.
  - Esperamos 10 us para que se cargue el condensador interno.
  - Cambiamos la línea a modo de entrada.
  - Contamos el tiempo que tarda en volver a ser un 0 lógico.
  - Apagamos los LEDs infrarrojos.

3. Aplicación
Desarrollo: Bluetooth

- Conectamos el pin TX del módulo al pin RX del Arduino.
- El 1 lógico en el Arduino está representado por 5 V, como en el módulo el nivel que representa al 1 es de 3.3 V necesitamos un divisor de tensión para conectar el TX del Arduino al RX del módulo.
- Para enviar y recibir información, escribimos y leemos en los pines TX y RX del Arduino en serie (funciones read y write).



3. Aplicación
Desarrollo: Construcción del terreno

- El terreno se recorrerá en varias filas.
- Cada fila tiene marcado su final con una tira negra (el robot avanzará hasta encontrar una).
- A lo largo de cada fila se encuentran las minas, representadas por tiras negras más pequeñas.

3. Aplicación
Desarrollo: Programa Arduino

- Elementos:
  1. Control del vehículo y gestión de sensores
  2. Comunicación con Android
  3. Comunicación con PC

3. Aplicación
Desarrollo: Programa Arduino

- Setup()
  - Inicializamos puerto serie (9600 Baud).
  - Obtenemos valor de referencia para el blanco.
  - Asociamos eventos generados por el giroscopio a una función encargada de obtener los valores.

3. Aplicación
Desarrollo: Programa Arduino

- Loop()
  - Si(primeraVez) -> Esperamos indicación para arrancar.
  - Comprobamos si hay datos disponibles en el puerto serie.
  - Comprobamos si hay objeto.
    - Mina: registramos el tiempo en el que la encontramos.
    - Control: registramos el tiempo en el que lo encontramos y consideramos la fila terminada.
      - Giramos (en base a los datos enviados por el Android) y avanzamos hasta encontrar otro control. Volvemos a girar y consideramos empezada una nueva línea.
  - Si hemos recorrido todas las filas, nos desconectamos del Android y esperamos una petición de datos desde el PC y enviamos parejas de Objeto-Tiempo al ordenador.

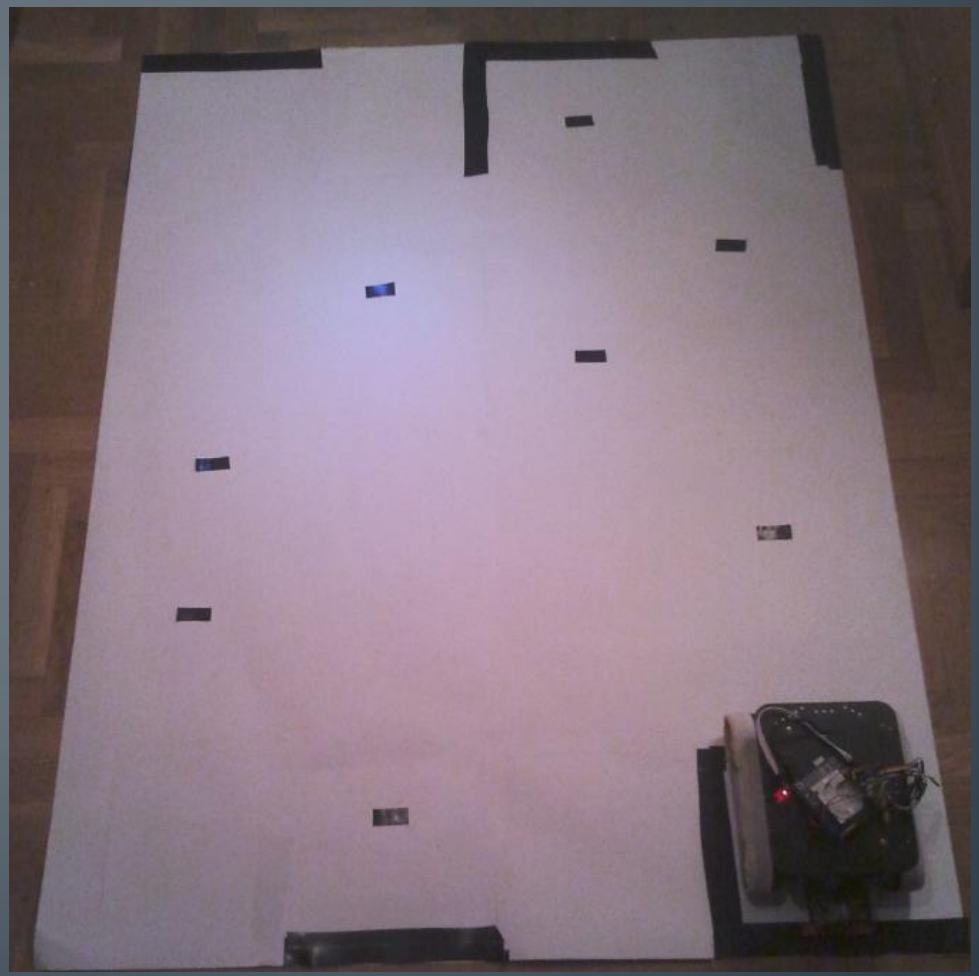
3. Aplicación
Desarrollo: Programa Arduino

- Comunicación con Android
  - Comprobamos si hay datos disponibles en el puerto y los almacenamos en un buffer hasta recibir el byte Ack, en cuyo procesamos lo recibido.
  - Si recibimos el byte Abord, vaciamos el buffer.

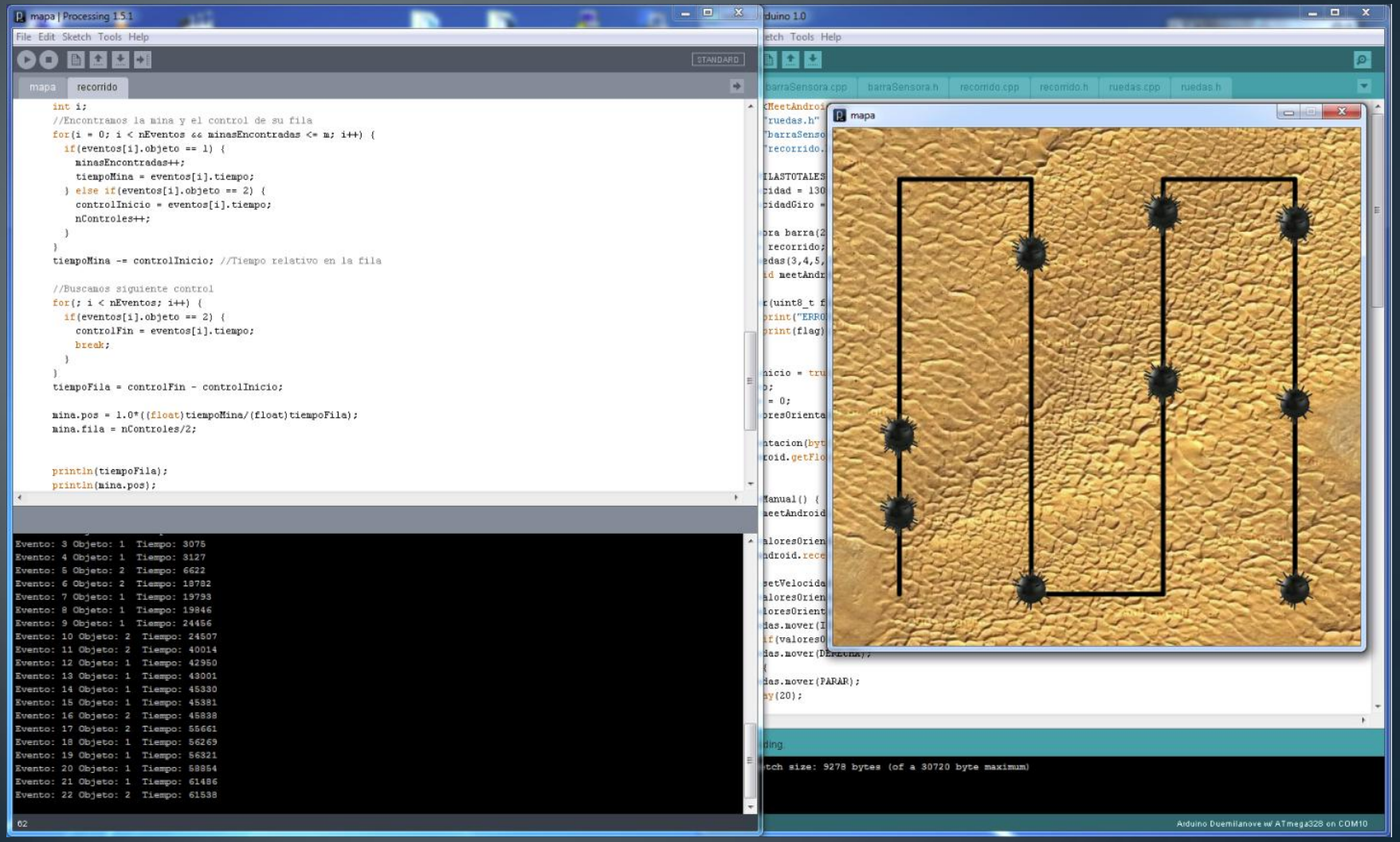
3. Aplicación
Desarrollo: Programa PC

- Setup()
  - Inicializamos configuración del puerto Bluetooth.
- Draw()
  - Dibujamos Fondo.
  - Enviamos petición de los datos y una vez recibidos los almacenamos.
  - Calculamos posición de las minas.
    - A partir del tiempo de inicio de fila y fin de fila, y el tiempo de la mina podemos determinar su posición.
  - Dibujamos las minas.

3. Aplicación
Resultado: Robot



3. Aplicación
Resultado: PC



4. Bibliografía

Bluetooth:

http://es.wikipedia.org/wiki/Bluetooth

http://es.wikipedia.org/wiki/Banda_ISM

Arduino:

http://arduino.cc/es/Tutorial/HomePage

http://arduino.cc/es/Reference/HomePage

Amarino:

http://www.amarino-toolkit.net/index.php/docs.html

Processing:

http://processing.org/about/

http://processing.org/reference/