



Universidad Nacional de La Matanza

**Departamento de Ingeniería e Investigaciones Tecnológicas
Ingeniería en Informática**

Sistemas Operativos Avanzados

Sistemas Embebidos - Arduino y Android

“SmartTrashCan”

Primer Cuatrimestre 2019 - Año 2019

Docentes:

- Lic. Graciela de Luca
- Ing. Waldo Valiente
- Ing. Esteban Carnuccio
- Ing. Mariano Volker
- Ing. Sebastián Barillaro
- Ing. Gerardo García

Integrantes:

- | | |
|--------------------|-----------------|
| • Ferrufino, Luis | DNI: 33.007.456 |
| • Flores, Gabriel | DNI: 35.957.073 |
| • Ojeda, Mariana | DNI: 35.902.635 |
| • Ponce, Brian | DNI: 36.946.425 |
| • Ramírez, Micaela | DNI: 39.432.357 |

Índice

Contenido

Índice.....	1
Introducción.....	3
Descripción General del Sistema.....	3
Alcance del Sistema	4
Sistema Embebido Arduino.....	4
Aplicación Android.....	4
Componentes	5
Software utilizado	5
Sistema embebido Arduino	5
Aplicación Android	5
Hardware utilizado.....	5
Sistema embebido Arduino	5
Aplicación Android	5
Circuito	6
Implementación	8
Sistema embebido y Android.....	8
Módulo Bluetooth.....	8
Sensor Infrarrojo Pasivo (PIR)	8
Sensor Ultrasónico	8
Sensor de Temperatura y Humedad.....	9
Sensor de Peso	9
Motor DC con Caja Reductora	9
Relé	9
LED	9
Micro Servo Motor	9
Aplicación Android.....	9
Descripción de los Componentes.....	12
Arduino Uno	12
Fuente de Alimentación	13
Módulo Bluetooth HC-05.....	13
Sensor PIR HC-SR501	13
Sensor Ultrasónico HC-SR04	14
En el pin echo el sensor devuelve al microcontrolador un pulso cuyo ancho es proporcional al tiempo que tarda el sonido en viajar del transductor al obstáculo y luego de vuelta al módulo.	14
Sensor de Humedad y Temperatura DHT11	14
Módulo de Celda de Carga HX711 de 5kg.....	14

Módulo de Relé con 5V.....	14
Motor DC de 3V a 6V con Caja Reductora.....	14
Micro Servo Motor SG90	14
LED	14
Diagramas	15
Diagrama Lógico	15
Diagrama Funcional	15
Diagrama de Hardware	16
Vista frontal.....	16
Vista trasera	17
Diagrama de Software.....	18

Introducción

A lo largo de este informe se explica el proyecto elegido por nuestro grupo para la materia Sistemas Operativos Avanzados, en el que pudimos experimentar en el mundo de los sistemas embebidos y de las aplicaciones mobile (Android).

Primero se describe en detalle la idea y la motivación por la que elegimos este proyecto y luego se explicarán los componentes que fueron utilizados. Por último, se muestran los diagramas que explican cómo está modelado.

Descripción General del Sistema

SmartTrashCan es un cesto de basura inteligente que permite almacenar residuos, manteniendo ordenado y limpio nuestro ambiente, ya sea en el ámbito laboral, de estudio o simplemente en el hogar, con la facilidad que nos brinda de no tener que levantar la tapa de forma manual, ya que detecta la presencia de la persona que arroja basura.

Aparte de la función anteriormente mencionada, la bolsa que se encuentra en su interior se cerrará automáticamente frente a diferentes tipos de detonantes: que alcance la capacidad máxima que soporta el cesto, que posea niveles de descomposición peligrosos para la salud o también, desde una aplicación Mobile.

Además de poder cerrar la bolsa desde la aplicación, también se podrá fijar un límite de peso máximo que al alcanzar, hará que la bolsa se cierre automáticamente. A modo de información, será posible consultar el peso del contenido del cesto o el estado de los residuos (considerando factores como humedad y peso).

Por último, se contará con un “Modo Juego”, en este modo la tapa subirá y bajará constantemente mientras que se intenta embocar objetos (no basura, ya que podrían caer en el piso), hasta que se agote el tiempo.

Alcance del Sistema

Sistema Embebido Arduino

Para el funcionamiento del cesto, se empleó un sistema embebido Arduino.

Cuando se detecta la presencia de una persona, la tapa se levantara automáticamente para permitir el ingreso de la basura que arrojara. La tapa permanece en este estado por un breve período de tiempo. Esto se logra mediante un sensor infrarrojo pasivo (PIR).

En el interior del cesto, se utilizó un sensor Ultrasónico para medir el volumen de la basura contenida, un Micro Servo Motor para el movimiento de la tapa; un sensor de humedad y temperatura que estará constantemente monitoreando los niveles de humedad y temperatura del contenido.

Cuando la distancia medida por el sensor Ultrasónico se reduzca a cierto valor establecido con anterioridad, o cuando se llegue al peso que se indicó como máximo, o cuando se le indicara desde la aplicación, se activara un relé que accionara un Motorreductor de Doble Eje para que la bolsa se cierre y esté lista para ser retirada por el usuario.

Para la comunicación con la aplicación Mobile se utilizó un sensor Bluetooth para el intercambio de datos.

Aplicación Android

La aplicación Mobile para Android es el sistema que provee la comunicación del smartphone con el cesto, la que se establece mediante el uso del Bluetooth. La aplicación tiene una interfaz sencilla con el objetivo de que sea fácil de usar para el usuario. Da la bienvenida, muestra los dispositivos emparejados al smartphone, y luego muestra un menú indicando todas las acciones que podemos hacer mediante el uso de la aplicación:

- Obtener peso actual
- Configurar peso máximo
- Activar modo juego
- Cerrar la bolsa
- Indicar que realicé el cambio de la bolsa
- Ver información sobre el estado de los residuos en el cesto

Hace uso de tres sensores del Framework de Android: sensor de proximidad, acelerómetro y sensor de luz ambiente. Es necesario activar estos sensores como es indicado por la aplicación para el envío de ciertas señales al embebido.

Componentes

Software utilizado

Sistema embebido Arduino

- IDE Arduino Genuino 1.8.9 (para el desarrollo del sistema embebido para la placa Arduino Uno).
- Fritzing (para el diseño de los diagramas esquemáticos de los circuitos electrónicos).

Aplicación Android

- IDE Android Studio 3.2.1 (para el desarrollo de la aplicación Mobile para Android).

Hardware utilizado

Sistema embebido Arduino

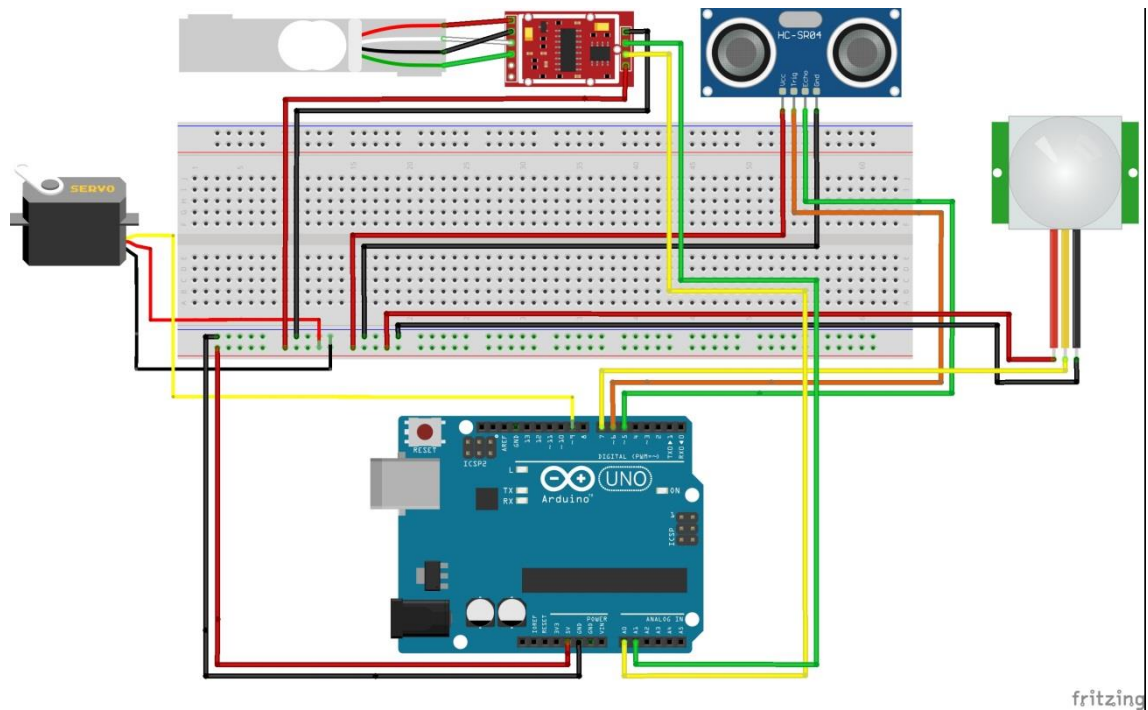
- 1 Placa Arduino Uno R3 Atmega328P
- 1 Cable USB puerto serial Arduino UNO 30cm
- 1 Protoboard
- 1 Módulo Bluetooth HC-05
- 1 Micro Servo Motor Tower Pro SG90 1.2kg
- 1 Sensor Infrarrojo Pasivo PIR HC-SR501
- 1 Sensor Ultrasónico HC-SR04
- 1 Sensor de Humedad y Temperatura DHT11
- 1 Módulo Relé 5v con Bornera de 250v
- 1 Motor DC de 3V a 5V con Caja Reductora
- 1 Celda de Carga HX711 de 10kg
- 1 Fuente transformador de 5V 1a
- 2 Resistencia 1K Ohms 1/4W
- 1 Led 5MM de 0.6 W
- 1 Bolsa con hilo

Aplicación Android

- Smartphone con sistema operativo Android
- Sensor de proximidad
- Acelerómetro
- Sensor de luz ambiente

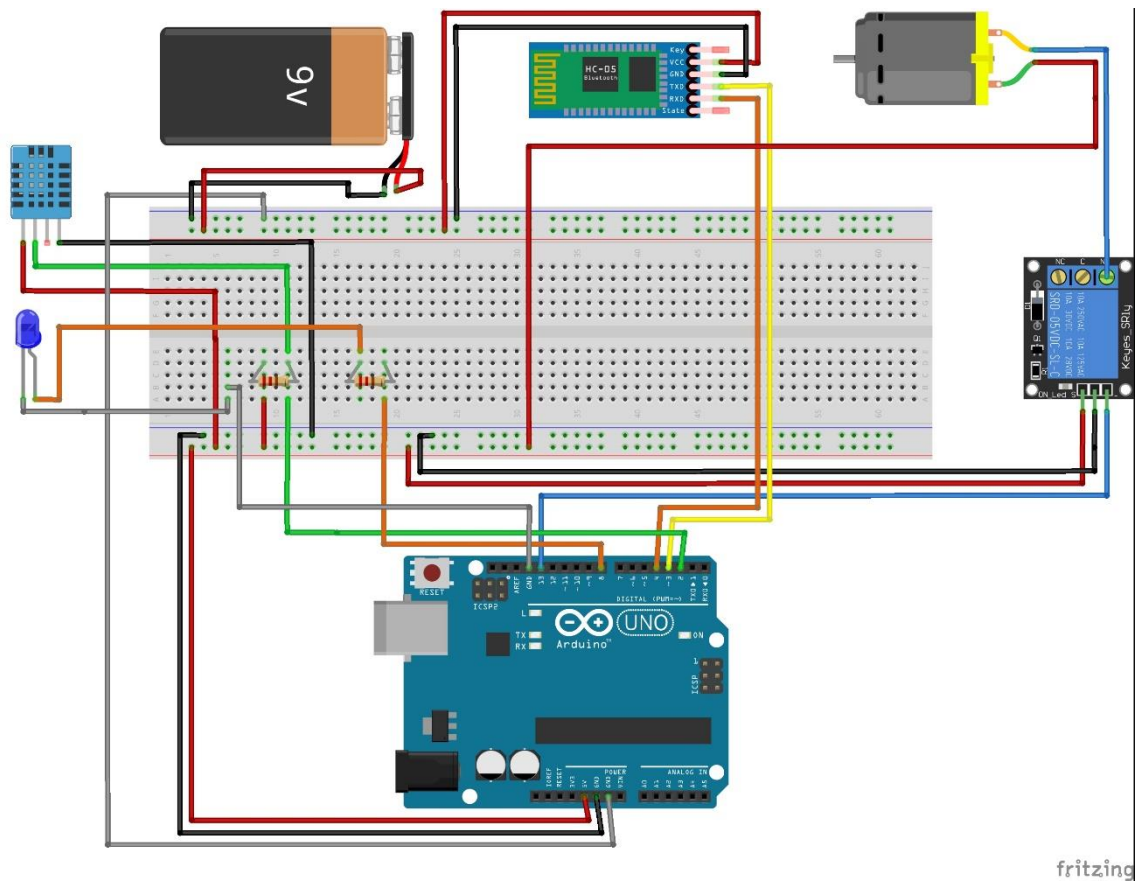
Circuito

Para realizar el diseño del circuito, se utilizó el Software Fritzing, una herramienta Open-Source. Para una mejor visualización se han separado en dos partes el circuito, y en cada una se ha incluido tanto la Placa Arduino como la Protoboard.



Componentes agregados:

- Sensor Infrarrojo PIR
- Sensor Ultrasónico
- Micro Servo Motor
- Celda de Carga + Acondicionador de señal



Componentes agregados:

- Módulo Relé
- Motor DC con Caja Reductora
- Módulo Bluetooth
- Celda de Carga + Acondicionador de señal
- Sensor de Humedad y Temperatura
- 1 Led
- 2 Resistencias

Implementación

Sistema embebido y Android

Módulo Bluetooth

Haciendo uso de la librería “SoftwareSerial.h”, desarrollamos la comunicación del módulo Bluetooth integrado en el Arduino con la aplicación Android. Es una pieza clave en el producto, ya que, a partir de dicha comunicación, se comparten datos que, en base a qué valores tengan, se realizarán distintas acciones (tanto en el sistema embebido Arduino como en la aplicación Android).

Se utilizan las siguientes funciones:

- **SoftwareSerial**: crea una nueva instancia del objeto SoftwareSerial.
- **available**: obtiene el número de bytes disponibles para la lectura de un puerto serie software.
- **begin**: se configura la velocidad en baudios para la comunicación serie.
- **read**: devuelve un carácter que se recibió en el pin RX del puerto serie de software.
- **write**: envía los comandos al dispositivo Android.

Desde el lado de la aplicación Android, se realizan tanto recepciones de datos desde el módulo bluetooth como el envío.

- Obtención del peso actual del cesto: se envía una señal para pedirle al embebido que envíe su peso y se realiza la lectura del valor enviado desde el embebido al Android, mostrándolo posteriormente.
- Configuración del peso máximo: envía al embebido una señal para indicar un peso máximo predefinido
- Activación del modo juego: se envía una señal al embebido para que comience el modo juego
- Cerrado de la bolsa: se envía una señal para que se cierre la bolsa, sin importar el estado actual de los residuos
- Cambiado de bolsa: se envía una señal para indicar que se realizó el cambio de la bolsa, tras esta señal, se inicializan variables para indicar que el tacho ya está operativo
- Información de residuos: envía una señal al embebido para pedir la información, el embebido se encarga de analizar distintos sensores para luego enviar al Android una cadena indicando el estado actual.

Sensor Infrarrojo Pasivo (PIR)

Para su implementación, a través de la función **digitalRead**, se puede saber si el sensor detecta la presencia de una persona. Se le debe configurar la sensibilidad tanto de distancia como de tiempo. Lo configuramos para que tenga un buen alcance para brindar más comodidades a las personas que lo usen. Fue necesario quitar el lente de Fresnel que tiene incluido para reducir su rango de detección y que solo detecte cuando una persona se acerca de frente. El sensor se encuentra ubicado junto a la tapa del cesto.

Sensor Ultrasónico

Para su implementación, desde el componente transmisor se emite una onda ultrasónica que, al chocar contra la basura, devolverá la onda reflejada que ingresará

por el componente receptor, obteniendo así el tiempo que tardo a través de la función **pulseIn**, de esta forma podemos determinar la distancia entre la tapa del cesto y la altura de la basura. El sensor se encuentra ubicado debajo de la tapa apuntando hacia el piso.

Sensor de Temperatura y Humedad

Para su implementación, se utilizaron las librerías “DHT.h” y “DHT_U.h”, de AdaFruit, que permiten la abstracción del protocolo propio utilizado en la comunicación entre el Sistema Embebido Arduino y el sensor de Temperatura y Humedad. El sensor es utilizado para obtener, desde la aplicación Mobile el estado de descomposición de la basura dentro del cesto.

Sensor de Peso

Para su implementación, se utilizó la librería “Adafruit_Sensor.h”

Motor DC con Caja Reductora

El motor es utilizado para cerrar la bolsa que contiene la basura, mediante el accionar del relé.

Relé

El relé es utilizado para accionar el motor. A través de la función **digitalWrite**, se pone en movimiento (HIGH) o se detiene al motor (LOW).

LED

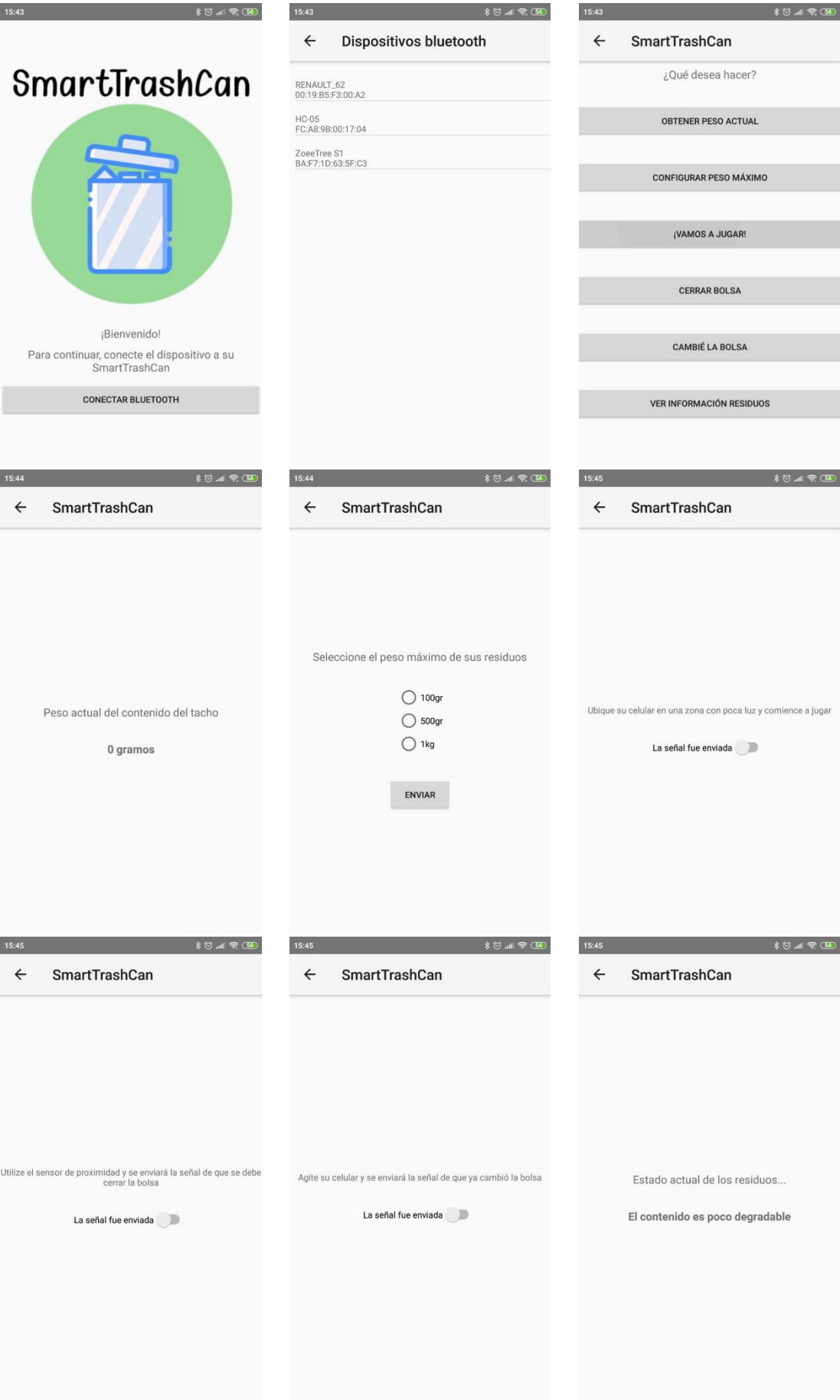
El led es utilizado para informar diferentes estados, por un lado, cuando se encuentra operando con normalidad el led permanece encendido, cuando la tapa se encuentra abierta esperando que se retire la basura el led permanece apagado y en el modo juego titila constantemente.

Micro Servo Motor

Para su implementación se utilizó la librería “Servo.h”, con el método **write** de la librería se lo activará y abrirá la tapa, o cerrará según corresponda. Este método recibe un número entero entre 0 y 90 que corresponde con el ángulo al cual se desplazará el eje.

Aplicación Android

A continuación, se muestran las activities que forman la aplicación:



Los sensores involucrados son:

- Acelerómetro

El sensor TYPE_ACCELEROMETER mide la aceleración aplicada al dispositivo y sus propiedades se resumen en la siguiente tabla:

Sensor	Tipo	Datos de SensorEvent (m/s ²)	Descripción
Acelerómetro	TYPE_ACCELEROMETER	values[0] values[1] values[2]	Aceleración en el eje x Aceleración en el eje y Aceleración en el eje z

Cuando se detecta un shake, se envía al embebido la señal de que la bolsa llena/cerrada ya fue cambiada.

- Sensor de proximidad

El sensor TYPE_PROXIMITY proporciona la distancia entre el dispositivo y otro objeto y puede por ejemplo usarse para determinar si la persona lo sostiene cerca de su cara al hacer una llamada o simplemente pasa su mano cerca de él. Sus propiedades se resumen en la siguiente tabla:

Sensor	Tipo	Datos de SensorEvent	Descripción
Proximidad	TYPE_PROXIMITY	values[0]	Distancia de un objeto en cm. Algunos sensores de proximidad solo declaran un valor booleano para indicar si el objeto se encuentra lo suficientemente cerca.

Cuando detectamos que el sensor de proximidad del smartphone está tapado, se envía al embebido la señal de cerrar la bolsa.

- Sensor de luz ambiente

El sensor TYPE_LIGHT proporciona la luz ambiente en unidad de medida lux.

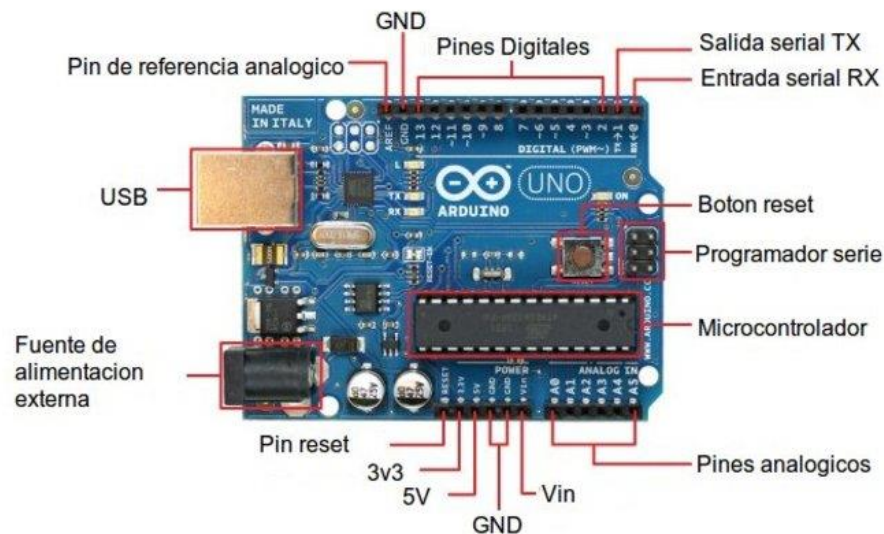
Cuando se detecta que hay poca luz ambiente, se envía al embebido la señal de que inicie el modo juego.

El modo de uso de la aplicación es muy sencillo:

1. Previamente, se debe emparejar el tachó con el dispositivo Android por medio de los ajustes del propio sistema.
2. Abrir la aplicación SmartTrashCan
3. Presionar el botón Conectar Bluetooth. Si no se tiene activado el bluetooth la aplicación pide encenderlo, luego muestra los dispositivos emparejados.
4. Presionar de la lista aquella que sea el tachó que desee controlar.
5. Realizar alguna acción de las indicadas en el menú.

Descripción de los Componentes

Arduino Uno



Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de sistemas embebidos a los que la comunidad de creadores puede darles diferentes tipos de uso. Esto es, en gran parte, gracias a la alta compatibilidad con un vasto conjunto de sensores y actuadores de los más variados tipos.

Esta, al igual que la mayoría, puede ser alimentada de varias formas, con un cable USB conectado al ordenador o con una fuente externa. El lenguaje de programación del microcontrolador está basado en C y C++.

Características técnicas:

Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Digital pines I/O	14 (de los cuales 6 proporcionan una salida PWM)
PWM digital pines I/O	6
Pines de entrada analógica	6
Corriente DC por Pin I/O	20mA
Corriente DC para Pin 3.3V	60mA
Memoria flash	32KB ATmega328P de los que 0,5 KB son utilizados por el gestor de arranque.
SRAM	2KB ATmega328P
EEPROM	1KB ATmega328P
Velocidad de reloj	16 MHz
Longitud	68,6 mm
Anchura	53,4 mm

Peso	25 g
------	------

Fuente de Alimentación



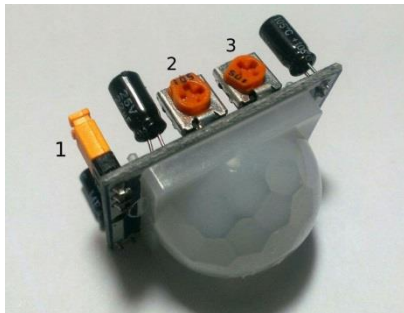
La fuente de alimentación es el dispositivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta. En nuestro caso, usamos una fuente de 5v, para alimentar la placa Arduino, junto con los sensores, actuadores, y el módulo bluetooth, para hacer al embebido independiente de tener que estar siempre conectado a una computadora y poderlo alimentar directamente con un tomacorriente convencional.

Módulo Bluetooth HC-05



Es el módulo que usamos para conectar el sistema embebido a un celular inalámbricamente. Se configura mediante comandos AT y tiene la posibilidad de hacerlo funcionar tanto en modo maestro como esclavo. Esto quiere decir que además de recibir conexiones desde una PC o celular, también es capaz de generar conexiones hacia otros dispositivos bluetooth. El módulo Bluetooth HC-05 puede alimentarse con una tensión de entre 3.3 y 6V (normalmente 5V).

Sensor PIR HC-SR501



El sensor incorpora 2 potenciómetros y un jumper que nos permiten modificar su comportamiento y adaptarlo a nuestras necesidades: precisión de detección, tiempo de reactivación, comportamiento ante detecciones repetitivas.

Los elementos de ajuste son los siguientes:

- **Selector de modo:** nos permite cambiar entre el modo de funcionamiento continuo o el modo de repetición. En modo continuo, si el sensor detecta movimiento de manera continuada mantendrá una señal continua. En el modo de repetición, el sensor se activará al detectar movimiento y volverá luego a su estado normal.
- **Ajuste de sensibilidad:** aumenta o disminuye la sensibilidad del sensor, con ello podemos ajustar la distancia a la que se activará y/o la cantidad de movimiento necesario para activar el sensor (por ejemplo, para distinguir una persona de una mascota).
- **Ajuste del temporizador:** aumenta o disminuye el tiempo que se activará el sensor una vez detecte presencia, el rango va aproximadamente desde unos 3 segundos hasta unos 5 minutos. Si el sensor se encuentra en modo continuo, el tiempo de activación será como mínimo el ajustado, no existiendo máximo si el sensor detecta continuamente presencia mientras se encuentra activado.

Sensor Ultrasónico HC-SR04



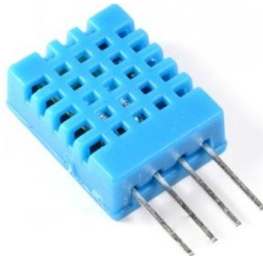
El sensor HC-SR04 es un módulo que incorpora un par de transductores de ultrasonido que se utilizan de manera conjunta para determinar la distancia del sensor con un objeto colocado enfrente de este. Un transductor emite una “ráfaga” de ultrasonido y el otro capta el rebote de dicha onda. El tiempo que tarda la onda sonora en ir y regresar a un objeto puede utilizarse para conocer la distancia que existe entre el origen del sonido y el objeto.

La interfaz del sensor HC-SR04 y Arduino se logra mediante 2 pines digitales: el pin de disparo (trigger) y eco (echo). La función de cada uno de estos pines es la siguiente:

El pin trigger recibe un pulso de habilitación de parte del microcontrolador, mediante el cual se le indica al módulo que comience a realizar la medición de distancia.

En el pin echo el sensor devuelve al microcontrolador un pulso cuyo ancho es proporcional al tiempo que tarda el sonido en viajar del transductor al obstáculo y luego de vuelta al módulo.

Sensor de Humedad y Temperatura DHT11



Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante y solo un pin para la lectura de los datos. Tal vez la desventaja de estos es la velocidad de las lecturas y el tiempo que hay que esperar para tomar nuevas lecturas (nueva lectura después de 2 segundos), pero esto no es tan importante puesto que la Temperatura y Humedad son variables que no cambian muy rápido en el tiempo.

No tenemos que confundirnos entre analógico y digital. Aunque lo conectemos a un pin digital, se trata de un dispositivo analógico. Dentro del propio dispositivo se hace la conversión entre analógico y digital.

Por lo tanto, partimos de una señal analógica que luego es convertida en formato digital y se enviará al microcontrolador.

Módulo de Celda de Carga HX711 de 5kg

Módulo de Relé con 5V

Motor DC de 3V a 6V con Caja Reductora

Micro Servo Motor SG90

LED

Diagramas

Diagrama Lógico

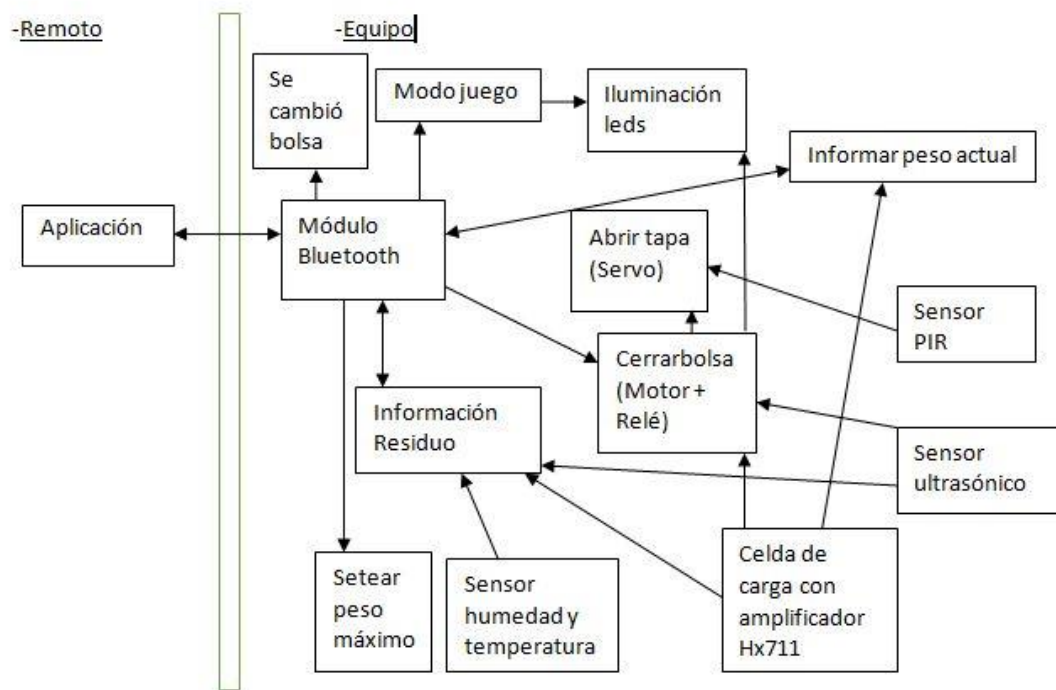


Diagrama Funcional

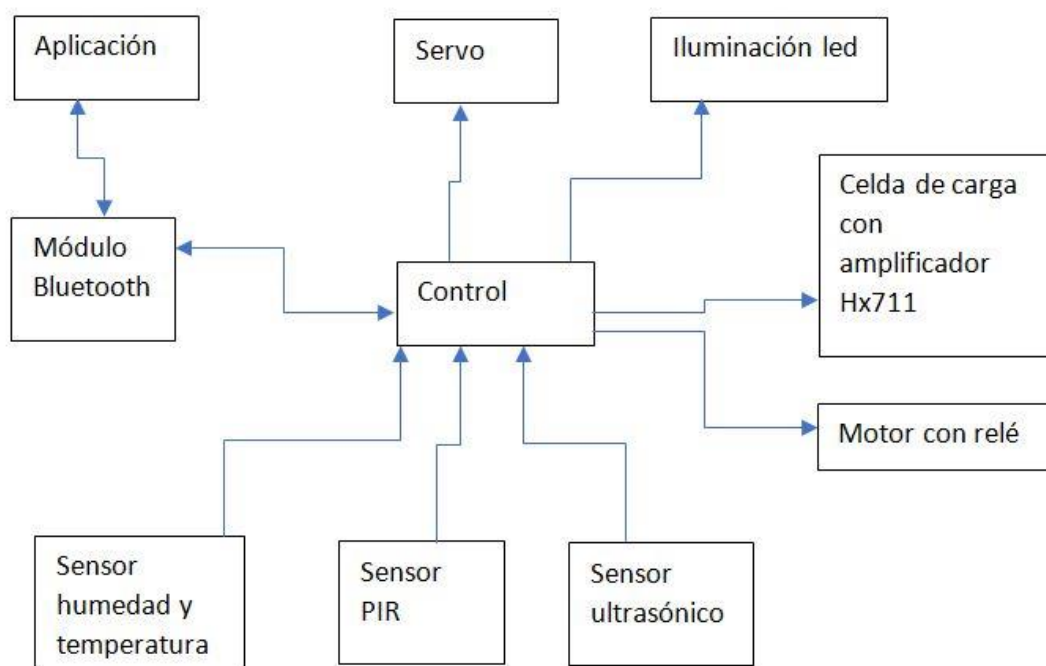
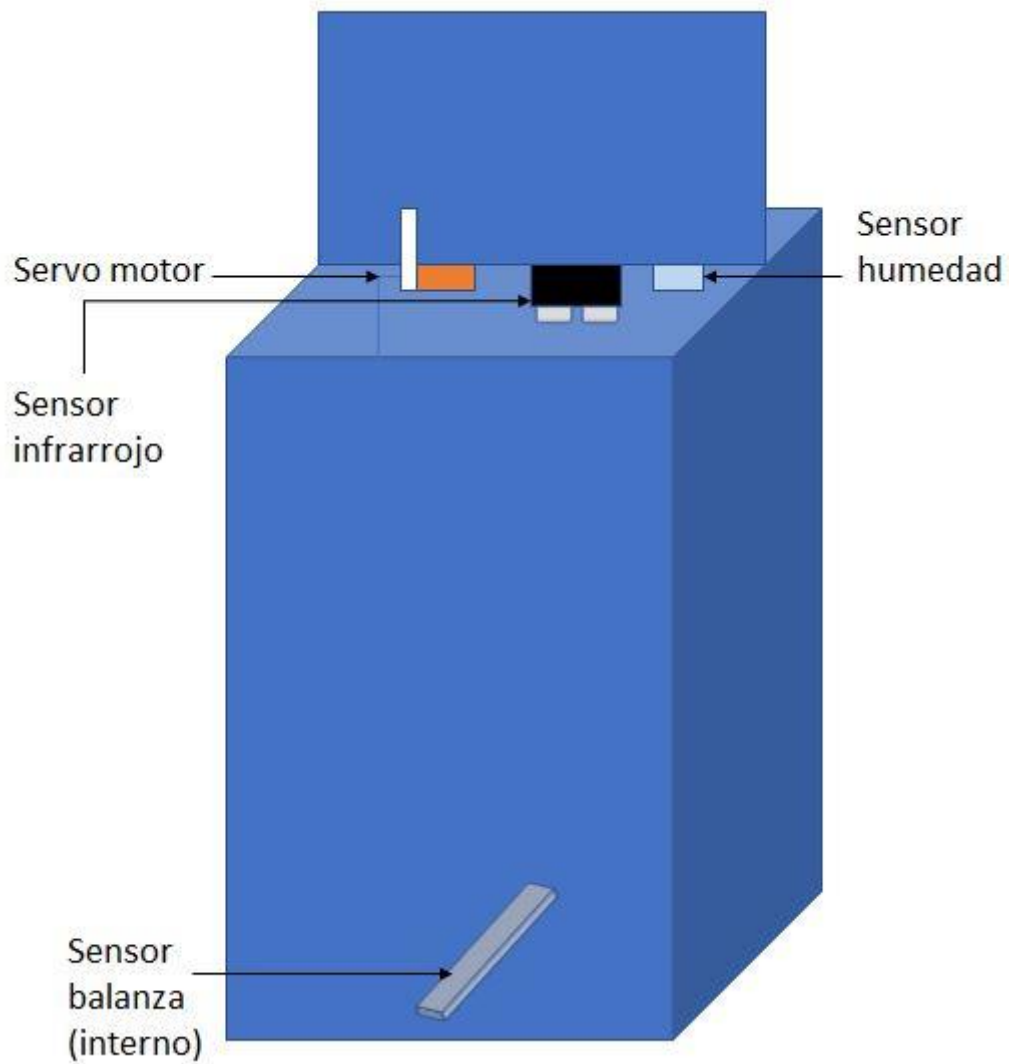


Diagrama de Hardware

Vista frontal



Vista trasera

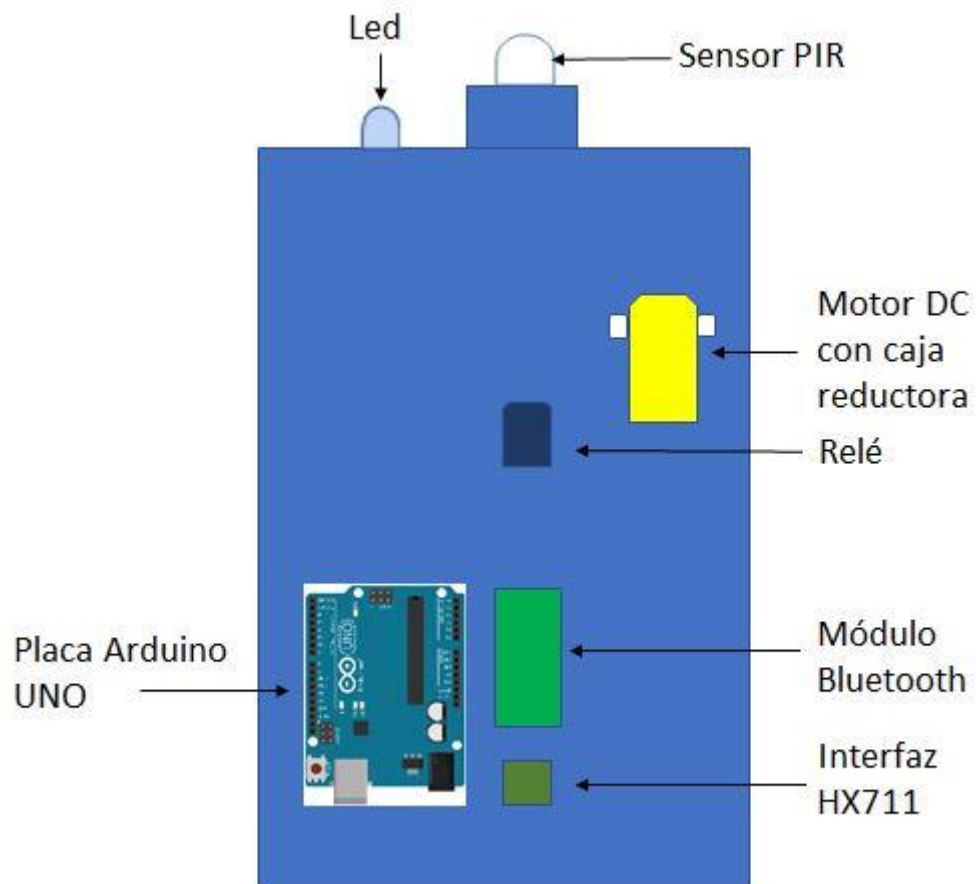


Diagrama de Software

