



*Universidad Nacional de La Matanza*

## **DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS**

### **SISTEMAS OPERATIVOS AVANZADOS**

**Sistemas Embebidos (IoT) y Android**

Comisión: Miércoles Noche

Ciclo Lectivo: 2º Cuatrimestre 2017

Docentes:

- Lic. De Luca Graciela
- Ing. Valente Waldo
- Ing. Carnuccio Esteban
- Ing. Volker Mariano
- Ing. Garcia Gerardo

Alumnos:

- |                     |                 |
|---------------------|-----------------|
| • Menendez Leonel   | DNI: 38.788.607 |
| • Raimondo Pablo    | DNI: 38.991.632 |
| • Ruttimann Hernán  | DNI: 38.614.315 |
| • Saccella Claudio  | DNI: 39.214.529 |
| • Scarpione Lautaro | DNI: 38.464.175 |
| • Vespoli Nicolás   | DNI: 37.925.580 |

INDICE:

**OBJETIVO:**

<b>ALCANCE:</b>	3
<b>HARDWARE UTILIZADO:</b>	4
MATERIALES UTILIZADOS:	4
SENSORES:	4
ACTUADORES:	4
SENSORES ANDROID:	4
<b>SOFTWARE UTILIZADO:</b>	4
<b>DETALLES TÉCNICOS DE LOS SENSORES UTILIZADOS:</b>	4
Sensor de temperatura LM35:	5
Sensor de gases MQ2:	5
Sensor de flama (Flame Sensor):	6
Barrera infrarroja:	6
Detector del nivel de agua:	7
<b>CONEXIONADO ELÉCTRICO:</b>	8
<b>DETALLES TÉCNICOS DE LOS ACTUADORES UTILIZADOS:</b>	9
<b>CIRCUITO REALIZADO:</b>	9
<b>APLICACIÓN ANDROID (FireBird):</b>	10

## OBJETIVO:

Desarrollar un sistema embebido (SE) del aspecto de una pajarera que se encargará de la detección de un principio de incendio mediante la obtención de datos del ambiente (llama, humo y temperatura).

En caso de detectar el incendio, el sistema reaccionará expulsando agua hacia abajo para intentar sofocar el fuego, además de encender una alarma tanto sonora como lumínica.

Como agregado, el sistema también proveerá el suministro de alimentación y bebida para aves de manera automatizada.

Desde la aplicación el usuario de FireBird podrá también informarse de los niveles de comida (Bajo, Medio o Alto) y agua restantes en los dispensers (Bajo o Alto), como así también de las mediciones de los sensores antes descritos.

El sistema embebido (pajarera) estará dispuesto a una altura de aproximadamente 5 metros sostenido de un árbol. Siendo como posibles ubicaciones bosques, campos, etc.

## ALCANCE:

- Estimar mediante mediciones de sensores el principio de un incendio.
- Expulsar agua en caso de detectar un incendio.
- Exender alimento para aves automáticamente.
- Dispensador de agua.
- Aviso sonoro y lumínico en caso de incendio.
- Informar las mediciones de los sensores.

## HARDWARE UTILIZADO:

- MATERIALES UTILIZADOS:
  - Placa Arduino UNO (14 entradas/salidas digitales, de las cuales 6 pueden ser PWM, y 6 entradas analógicas).
  - Módulo Bluetooth HC-05 (Maestro-Esclavo).
  
- SENSORES:
  - 1 sensor de flama (Flame Sensor).
  - 1 sensor de gases MQ2.
  - 1 sensor de temperatura LM35.
  - 3 barreras infrarrojas (2 nivel de comida en el dispenser y 1 en el plato).
  - 2 detectores del nivel de agua (1 para el dispenser de la bebida y 1 para el dispenser anti incendios).
  
- ACTUADORES:
  - 1 buzzer 5V (alarma sonora).
  - 1 LED (alarma lumínica).
  - 1 servomotor SG90 (dispensador de comida).
  - 1 sapito 12V (expulsador de agua en caso de detectarse un incendio).
  
- SENSORES ANDROID:
  - Acelerómetro.
  - Giroscopio.
  - Sensor de proximidad.

## SOFTWARE UTILIZADO:

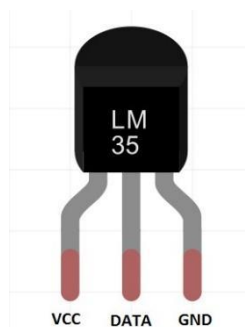
- Android Studio.
- Arduino 1.8.5.
- Versión de Android 5.0 (mínimo).
- Fritzing (Diseño de circuitos).

## DETALLES TÉCNICOS DE LOS SENSORES UTILIZADOS:

- **Sensor de temperatura LM35:**

- Tipo de señal: Analógica.
- Unidad de medida: Celsius.
- Rango de medición de temperatura: -55 a 150°C.
- Rango de medición garantizada: 0.5 a 25°C.
- Alimentación: 4 a 30V.

El LM35 es un sensor de temperatura con una precisión calibrada de 1 °C. No necesita circuitería externa ni ajustes de calibración para funcionar. Es ideal para aplicaciones remotas ya que consume menos de 60 mA de corriente con una salida de baja impedancia. La salida es lineal y cada grado Celcius equivale a 10 mV.



- **Sensor de gases MQ2:**

- Tipo de señal: Analógica.
- Tiempo de Respuesta:  $\leq 10s$ .
- Tiempo de recuperación:  $\leq 30s$ .
- Consumo: 800mW.
- Alimentación: 5V.
- Temperatura de trabajo: -20°C ~ +55°C
- Humedad:  $\leq 95\%$  RH.
- Rango de detección: 300 a 10000 ppm.
- Resistencia interna: 33 Ohm.

El sensor de gas analógico (MQ2) detecta la presencia de gas combustible y humo en concentraciones de 300 a 10.000 ppm. Incorpora una sencilla interfaz de tensión analógica que únicamente requiere un pin de entrada analógica del microcontrolador.

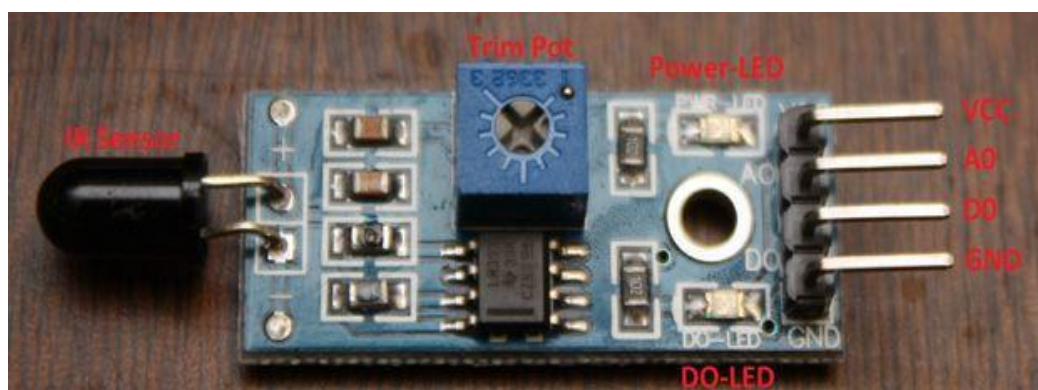
Este sensor es electroquímico y varía su resistencia cuando se expone a determinados gases, internamente posee un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia.



- **Sensor de flama (Flame Sensor):**

- Tipo de señal: Analógica/Digital (Nosotros usaremos la entrada analógica).
- Voltaje: 5V.
- Corriente: 20mA.
- Rango de ancho de banda espectral: 760 a 1100nm.
- Tiempo de respuesta: 15 $\mu$ s.
- Temperatura de trabajo: -25 a 85°C

Un detector de llama es un sensor diseñado para detectar y responder a la presencia de una llama o fuego, permitiendo la detección de llamas.



- **Barrera infrarroja:**

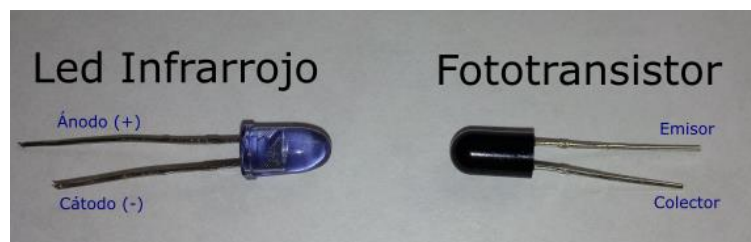
Compuesto por un LED IR (Infra Rojo) y un fototransistor:

- LED IR:

Este LED emite un tipo de radiación electromagnética llamada infrarroja, que es invisible para el ojo humano porque su longitud de onda es mayor a la del espectro visible.

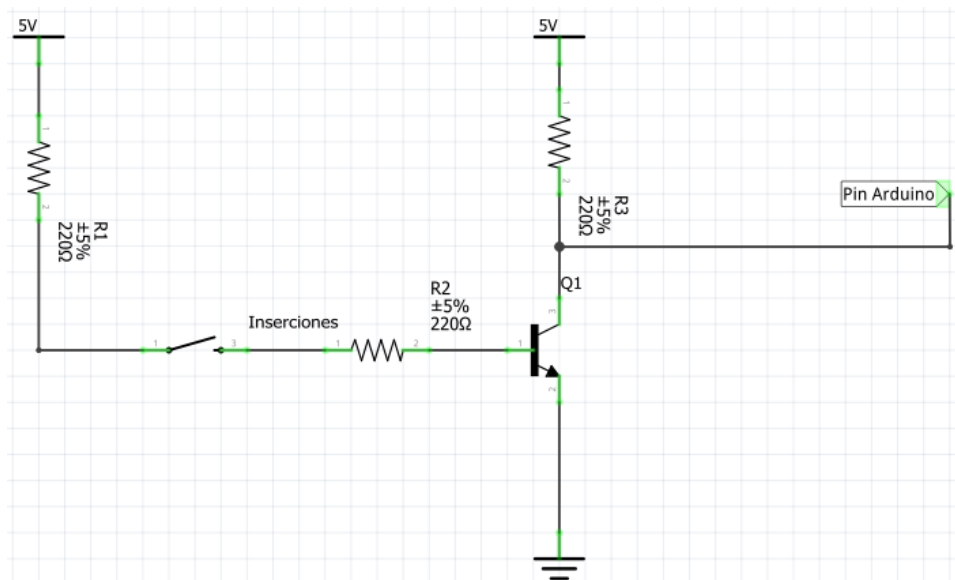
- Fototransistor:

Este dispositivo se diferencia de un transistor común en que su base ha sido sustituida por un cristal fotosensible que regula el flujo de corriente colector – emisor de acuerdo con la luz incidente sobre él (en nuestro caso luz infrarroja).

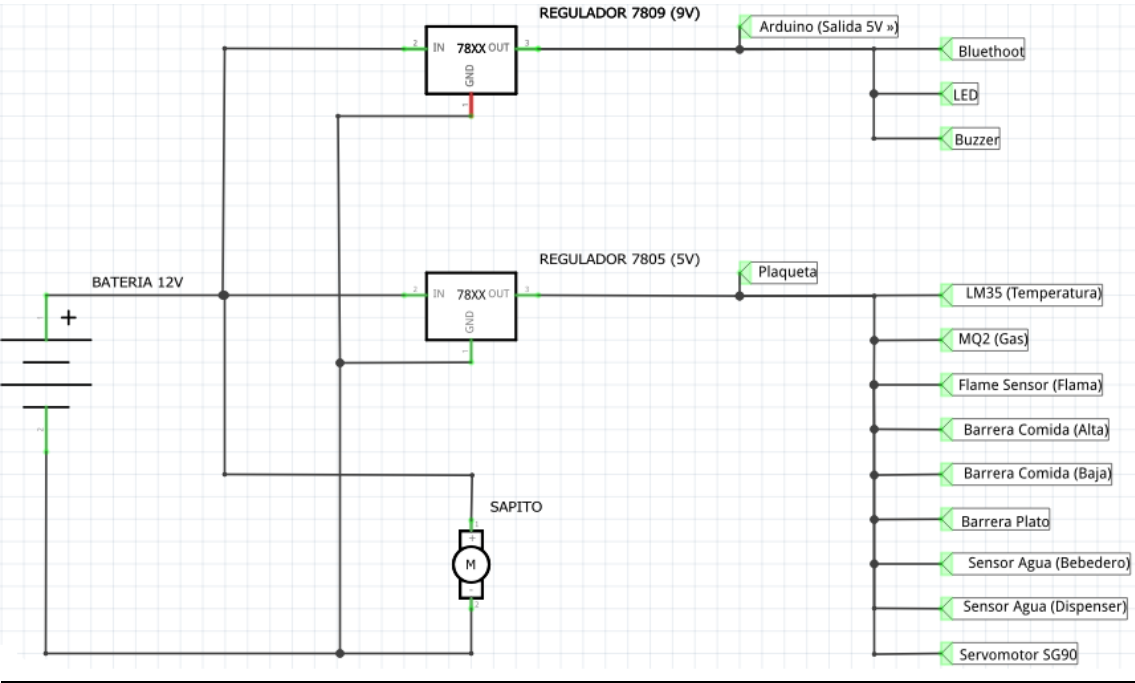


- **Detector del nivel de agua:**

Este sensor detectará el nivel de agua basándose en dos inserciones de un material conductor en el dispenser de agua. A partir de eso, detectará si el agua conecta estas dos inserciones funcionando como un “puente” que enviará una señal al Arduino de la siguiente manera:



CONEXIONADO ELÉCTRICO:





## DETALLES TÉCNICOS DE LOS ACTUADORES UTILIZADOS:

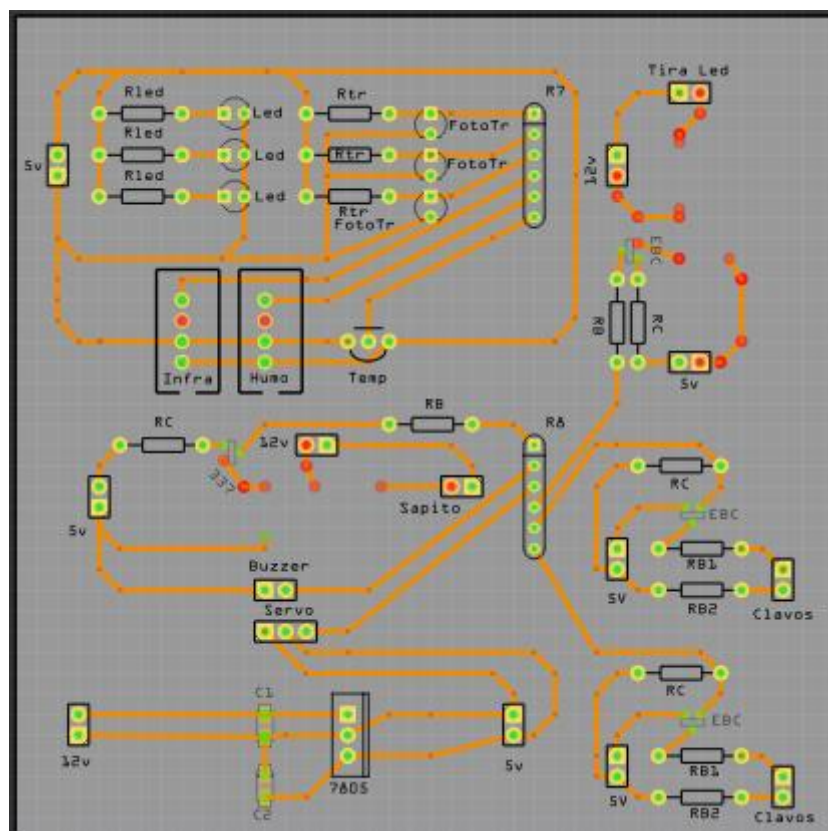
- **Servomotor SG90:**
  - Voltaje: 4.8V a 6V.
  - Peso: 14.7 gramos.
  - Torque: 2.5 Kg/cm.
  - Velocidad: 0.1 seg/60°.
  - Giro: 180° (-90° +90°).

El movimiento de este dispositivo se realiza mediante una señal PWM. Su posición de reposo es en el que se encontraba cuando se detuvo por última vez. El período del PWM es de 20 ms (50Hz). Mientras que los ciclos de trabajo que deben de usarse para realizar el giro del servomotor son los siguientes:

- 1 ms: -90°.
- 1.5 ms: 0°.
- 2 ms: +90°.

## CIRCUITO REALIZADO:

La plaqueta realizada es la siguiente:



## APLICACIÓN ANDROID (FireBird):

La aplicación Android tiene 2 Activities:

- Main Activity:

Esta activity se utiliza para realizar la conexión BT con el Arduino.

- Info Activity:

Esta activity posee 2 hilos que corren en background y 3 fragments. Las funcionalidades de los hilos son:

- Envía una solicitud al Arduino para conocer las mediciones de los sensores y espera la respuesta, guardando estos datos en la base de datos.
- Cada un tiempo dado almacena en la base de datos la última medición tomada (solo se almacenan las últimas X mediciones).

Los 3 fragments:

- Fragment Info:

En este fragment se muestran los niveles de flama, temperatura, humo, agua y comida actuales. Para esto hay un hilo. El que cada un tiempo dado, actualiza la pantalla tomando los datos de la base de datos.

- Fragment History:

En este fragment se muestra el historial de las mediciones tomadas por La Pajarera. Este fragment también contiene un hilo. El cual cada un tiempo dado, lee los datos históricos de la base de datos y actualiza la pantalla.

- Fragment Gestures:

En este fragment se muestra en detalle cuales son los gestos que reconocerá la aplicación y que utilidad tendrá cada uno. Estas funcionalidades solo servirán para testear a La Pajarera, ya sea la alarma sonora (buzzer), la alarma lumínica (tira de LED), y que el “sapito” expulse agua correctamente.

Los gestos antes descriptos son escuchados en el Info Activity, lo que significa que en cualquiera de los fragments pertenecientes a la activity serán escuchados.

Para estos gestos se utilizarán los siguientes sensores de Android:

- Acelerómetro: Alarma lumínica.

- Giroscopio: "Sapito".
- Proximidad: Alarma sonora.

### **Comunicación entre la aplicación y el Arduino:**

La comunicación se realizará mediante BT (HC-05).

La aplicación le enviará al Arduino un entero (int) representando un código, según el código recibido, el Arduino realizará una función distinta. Los códigos son:

- "50": La aplicación le pide al Arduino los valores de las mediciones de los sensores.
- "11": Si el Arduino recibe este código testeará la alarma lumínica.
- "22": Si el Arduino recibe este código testeará la alarma sonora.
- "33": Si el Arduino recibe este código testeará la correcta expulsión del agua.

En caso de recibir el código "50", el Arduino le enviará la respuesta por medio de un "string" en formato JSON.