

**UNIVERSIDAD NACIONAL DE LA MATANZA**

**DEPARTAMENTO DE INGENIERÍA**

**SISTEMAS OPERATIVOS AVANZADOS**

Trabajo práctico de Sistemas embebidos y Android - IoT

**Sistema de Invernadero Inteligente: “Smart Roots”**

INTEGRANTES:

* Blanco, Juan Manuel
* Di Giacomo, Gastón
* Rosenfeld, Diego
* Strficek, Ivo

**Comisión:** Miércoles Noche

**Año:** 2016 – Segundo cuatrimestre

**Introducción**

En el siguiente documento explicaremos el desarrollo del sistema realizado en la cátedra de Sistemas Operativos Avanzados de la Universidad Nacional de La Matanza. La consigna era crear un sistema embebido dentro de una placa Intel Galileo / Arduino que actúe como servidor, y conectarlo con un celular que actúe como cliente, aprovechando los sensores de ambos para brindar alguna funcionalidad particular.

Este equipo de trabajo se propuso desarrollar un sistema embebido dedicado a la auto regulación de factores climáticos en invernaderos o entornos de cultivo controlados. Para ello, ideamos un proyecto progresivo y escalable, que va evolucionando a su siguiente versión mediante el cumplimiento de hitos, que se pasaran a detallar a lo largo del documento.

**Entorno de Desarrollo**

Hardware Utilizado

**Para el sistema embebido:**

* 1 Placa Arduino UNO
* 1 Sensor de Temperatura LM 35
* 1 Sensor RTC (Real Time Clock)
* 2 Ventiladores
* 2 Relays (doble inversor) 5V, 0.5A
* 1 Cable USB – miniPlug
* 1 Cable Ethernet
* 1 Lámpara de luz blanca

**Para la aplicación Android:**

* Smartphone Samsung S3 con Android 4.1
* 1 Notebook con Procesador Intel® Core™ i3, 4GB RAM, 500 GB HDD
* 1 Cable USB – microUSB.

Software Utilizado

**Para el sistema embebido:**

* Putty SSH, Telnet and Login Client.
* Arduino IDE.
* Ccgetmac.

**Para la aplicación Android:**

* Android Studio IDE (última versión).

**Descripción del Proyecto**

Planteo general del sistema embebido

Realizamos un producto especializado en el monitoreo y control de sistemas de invernadero que cuenta con las siguientes características:

* Control y regulación de la temperatura ambiental, con informes a través de la aplicación Android.
* Control del ciclo lumínico de las lámparas, pudiendo simular las distintas estaciones (estableciendo un rango horario del día durante el cual la lámpara permanecerá encendida)
* Administración del sistema de ventilación (tomando las mediciones de la temperatura ambiental).

Sensor de Temperatura

Con una precisión calibrada de 1 ºC. Su rango de medición abarca desde -55 °C hasta 150 °C. La salida es lineal y cada grado Celsius equivale a 10 mV.

Sus características más relevantes son:

* La tensión de salida es proporcional a la temperatura.
* Tiene una precisión garantizada de 0.5 °C a 25 °C.
* Baja impedancia de salida.
* Baja corriente de alimentación (60 μA).
* Bajo coste.



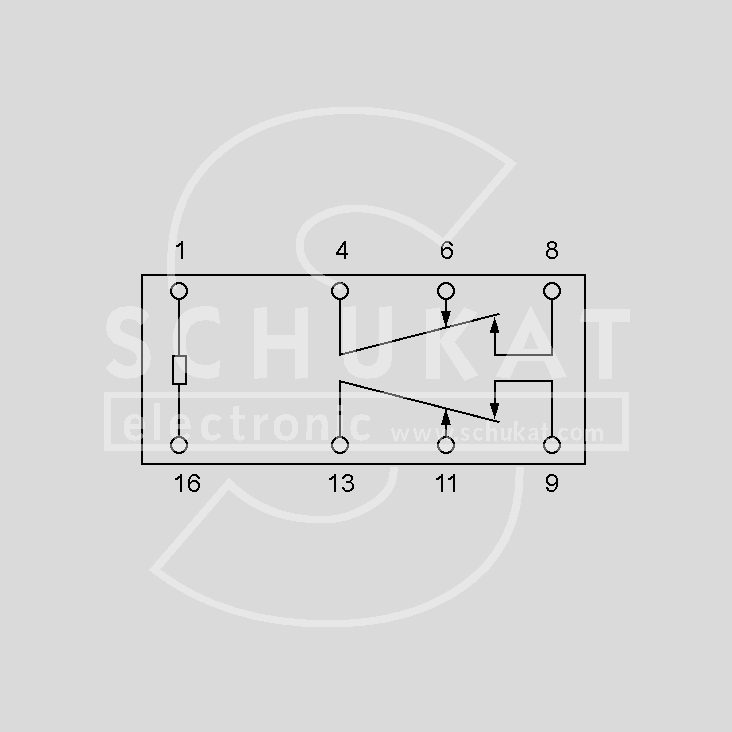
Para calcular la temperatura en grados Celsius, a partir del valor analógico obtenido desde la placa (en voltaje), se realiza el siguiente cálculo>

**((Value/1023)\*5) = [celcius]**

Relays

* Doble inversor
* 0.5A x 5V





Conexionado interno

Relay 1 (Control de Coolers)

Manda una señal de activación o desactivación según corresponda a los dos coolers encargados de disminuir la temperatura ambiente. Uno se encarga de ingresar aire frio del exterior y el otro extrae el aire caliente del interior.

Relay 2 (Control de Lámpara)

Este actuador hace de intermediario en el control de la lámpara emisora de calor. Su bobina se activa con 5V, mientras que la salida se conecta a 220V tensión necesaria para activar la lámpara.

Haciendo uso del sensor RTC, se establecerá un rango horario del día durante el cual la lámpara permanecerá encendida.

Hitos:

**Hito 0:** Construcción del modelo a escala del invernadero

*Hito alcanzado: 27/10/2016*

**Hito 1:** Obtener Mediciones de Temperatura.

*Hito alcanzado: 01/10/2016*

**Hito 2:** Informar las mediciones de temperatura.

*Hito alcanzado: 01/10/2016*

**Hito 3:** Accionar los relay para el control de los coolers, en función de la temperatura medida.

*Hito alcanzado: 08/10/2016*

**Hito 4:** Manejar el relay para el control de la lámpara.

*Hito alcanzado: 08/10/2016*

**Hito 5:** Creación de la Aplicación Android

*Hito alcanzado: 20/10/2016*

**Hito 6:** Lograr la conexión entre la placa Arduino y la Aplicación Android

*Hito alcanzado: 29/10/2016*

**Hito 7:** Comunicación mediante HTTP requests entre la placa Arduino y la Aplicación Android.

*Hito alcanzado: 2/11/2016*

Interacción entre la Placa y la aplicación Android

Para la interacción entre la placa y la aplicación Android decidimos utilizar una arquitectura de comunicación Cliente-Servidor mediante requests HTTP (REST / JSON), donde la placa Arduino actúa como servidor, y la aplicación Android se conecta como cliente. Los servicios que definimos para hacer efectiva la comunicación son:

* ***/temperatura***

Solicita a la placa la medición actual de la temperatura del invernadero, recibiendo como respuesta la temperatura en ºC.

* ***/luz\_on***

Envía una solicitud a la placa para encender la luz del invernadero, y recibe un mensaje de confirmación del encendido.

* ***/luz\_off***

Envía una solicitud a la placa para apagar la luz del invernadero, y recibe un mensaje de confirmación del apagado.

* ***/rango***

Envía dos valores a la placa para definir un nuevo rango de temperatura sobre el cual se accionaran los ventiladores (MIN,MAX). Es decir, los ventiladores se encenderán cuando la temperatura se encuentre por arriba del máximo, y se apagarán cuando estén por abajo del mínimo. Recibirá un mensaje de confirmación sobre dicha modificación.

Detalles de diseño y funcionamiento de la aplicación Android

* ***Funciones***
  + Consultar temperatura de invernadero
  + Encender ventiladores
  + Apagar ventiladores
  + Modificar el rango horario de iluminación del invernadero
* ***Sensores utilizados en la aplicación*** 
  + Micrófono: Mediante reconocimiento de voz, se puede hacer uso de algunas de las primeras 3 funciones de la aplicación.
  + Acelerómetro: Agitando el smartphone, se cierra la aplicación.
* ***Aclaraciones / Salvedades***
  + Para la conexión contra la placa Arduino, se hace uso de la clase de Java import “*net.URLConnection*”.
  + Para hacer uso de dicha clase e invocar su método “*openConnection*”, se implementa una “*AsyncTask*” (tarea asincrónica). Nos vimos obligados a realizar esto, ya que de otra forma la aplicación se cerraba abruptamente debido a un error del MainThread. Así, la función de conexión contra la placa se realiza de modo asincrónico y el MainThread se sigue ejecutado de manera paralela.
  + Para el reconocimiento de voz se hace uso de la API de android “*speech.RecognizerIntent*”, nativa de android. Al recibir una entrada de audio, nos devuelve un array de strings con las diferentes palabras que interpretó. Nosotros optamos por quedarnos con la primer palabra/frase de ese array, ya que es la definida con mayor “certidumbre” por la API. En caso de que la palabra que se desee enviar (por ejemplo: “temperatura”) no sea reconocida correctamente, la aplicación informará al respecto, pidiendo enviar la órden nuevamente.
  + Para utilizar el acelerómetro, utiizamos la clase “*hardware.SensorManager*” ye implementamos la clase “*ShakeDetector*” (extendida de “*SensorEventListener*”), definiendo allí los valores específicos de gravedad, tiempos y cantidad de “shakes” para que la aplicación se cierre cuando corresponda.
  + Por último, para mostrar una pantalla de inicio antes de la actividad principal de la aplicación, implementamos la clase “*SplashScreen*”, que se ejecuta como thread únicamente cuando se inicia la aplicación, dando lugar luego a la MainActivity.
* ***Capturas***

