

Lector de sensores arduino-android.

Programación de Sistemas - Grao en Ingeniería Informática 1ºQ

20/10/2014

Persona de contacto: Santiago García Lesta
<santiago.lesa@udc.es>
Oscar Blanco Novoa
<o.blanco@udc.es>

1 Introducción.

1.1 Objetivo:

El proyecto se centra en la recogida de datos de 1 o varias estaciones de sensores situadas en distintos puntos de la geográfica, con el objetivo de almacenar y estudiar los datos obtenidos.

El funcionamiento básico consiste en conectar el teléfono móvil a la estación de sensores controlada por arduino y establecer la conexión. Una vez hecho esto la estación de sensores enviará los datos almacenados a la aplicación y la mantendrá actualizada hasta que se desconecte el cable.

Además de los datos de los sensores se guardará información sobre la estación como la posición gps, y se podrá actualizar la configuración de la misma como la frecuencia de actualización de los sensores o el angulo de incidencia de el sensor de luminosidad (servo).

1.2 Motivación:

Dicha estación de sensores es funcional hoy en día pero carece de una interfaz adecuada para motorizar los datos obtenidos, además la posición de la estación puede ser un lugar de montaña sin cobertura, por lo tanto es interesante una aplicación para recoger los datos y almacenarlos.

1.2 Intereses:

Modificar el código actual de la estación para que envíe los datos a través de la entrada usb del dispositivo.

2 Análisis de requisitos.

2.1 Funcionalidades que implementaremos:

- Motorización de datos en tiempo real.
- Almacenaje de datos
- Visualización de un histórico
- Modificación de la configuración de la estación
- Comunicación vía wifi.
- Obtención de datos remotamente desde otro terminal.
- Guardar configuración de la aplicación.
- Adición de nuevos sensores desde el dispositivo android.

2.2 Prioridades y orden:

- 1.Comunicación desde el arduino hacia el dispositivo android. Dependencias()
- 2.Comunicación desde el dispositivo android hacia el arduino. Dependencias()
- 3.Refactorización del código de la estación de sensores. Dependencias()

4. Motirización de datos en tiempo real .Dependencias (1,2,3)
5. Modificación de la configuración de la estación. Dependencias (1.2,3,4)
6. Almacenaje de datos. Dependencias(4)
7. Visualización de un histórico(6)
8. Adición de nuevos sensores desde el dispositivo android. Dependencias(4)
9. Guardar configuración de la aplicación. Dependencias(8)
10. Obtención de datos remotamente desde otro terminal. Dependencias(4,6)
11. Comunicación vía wifi. Dependencias(1,2,3)

Básicos= 1,2,3,4.

En caso de acabar los puntos básicos antes de la fecha limite implementaremos los puntos 5, 6 y 7.

En caso de haber realizado los puntos anteriores realizaremos los puntos 8 y 9.

En el mejor de los casos implementaremos el punto 10 y el 11.

2.3 Responsabilidades:

Individuales:

Santiago García Lesta será responsable de los puntos: 1 2 6 7.

Oscar Blanco Novoa será responsable de los puntos: 3 5 8 9.

Colectivas:

Debido a la imposibilidad de trabajo en paralelo(una sola estación de sensores y posible complejidad) los siguientes puntos serán responsabilidad de los dos: 4 9 10.

3.1 Hitos:

10º semana: Funcionalidad básica terminada. (test de correcta visión de los datos).

11º semana: Funcionalidad básica más la opción de la configuración de la estación.

13º semana: Posibilidad de almacenaje y visualización de históricos.

14º semana: Añadir y eliminar de forma dinámica sensores y guardar configuración.

16º semana: Acceso remoto y comunicación wifi.

3.2 Posible planificación:

Conocimientos previos: experiencia con arduino, experiencia con las librerías de comunicación del arduino.

Se ha probado la comunicación con el arduino desde el dispositivo. La estación de sensores ha sido probada. Se debe testear la comunicación por wifi.

Posibles errores: problemas de capacidad por la librerías de comunicación en el arduino.

4.1 Diseño: Una única pantalla para visionar los datos en tiempo real y un botón para conectar con el dispositivo y opcionalmente gráficas.

4.2 Posible arquitectura:

Posiblemente con el avance del proyecto la arquitectura será adaptada a las necesidades encontradas.

Tendremos una actividad principal donde en una primera instancia visualizaremos los datos en tiempo real y tendremos la opción de conectarnos con el dispositivo.

Utilizaremos un broadcast receiver para gestionar la conexión usb.

En segundo plano tendremos una actividad que se ocupe de la comunicación con el arduino.

Desde la actividad principal lanzaremos una AsyncTask que se ocupe de pedir los datos al servicio y actualizar los datos en la pantalla de la actividad principal.

En la segunda entrega añadiremos una segunda actividad llamada desde la primera para gestionar la configuración de la estación base.

En la tercera entrega le añadiremos al servicio la posibilidad de guardar los datos que reciba en una base de datos y desde la actividad principal la opción de llamar a una tercera actividad para la visualización de dichos datos.

En la cuarta entrega añadiremos la opción en la actividad de añadir y eliminar sensores y hacer persistente dicha configuración.

En la quinta entrega añadiremos la opción de que el servicio se comuniqué vía wifi

Primera entrega

1 Funcionalidades.

Las funcionalidades implementadas hasta el momento son las siguientes.

- 1.Comunicación desde el arduino hacia el dispositivo android.
- 2.Comunicación desde el dispositivo android hacia el arduino.
- 3.Refactorización del código de la estación de sensores.
4. Motorización de datos en tiempo real.

2 Diseño.

Una única pantalla para visionar los datos en tiempo real, un botón para conectar con el dispositivo y un botón con estado para gestionar una luz en la estación. En caso de perder la conexión con la placa se pide automáticamente la autorización del usuario para reconectar.

3 Planificación.

Hemos aprovechado la existencia de la estación de sensores funcional y hemos modificado el código para hacerlo compatible con nuestra aplicación para android.

4 Problemas encontrados.

Se nos ha presentado el problema de que el hardware de la estación de sensores utiliza los pines 10-13 del arduino como salidas SPI, algo que no es compatible con nuestro arduino Leonardo ya que esta versión solo implementa la salida específica de SPI de la placa, lo que nos ha obligado a deshabilitar la pantalla LCD y uno de los sensores. Algo que intentaremos solventar en próximas iteraciones.

Otro de los problemas que nos ha retrasado ha sido el no poder depurar por usb mientras estamos obteniendo datos de la estación, al estar el puerto USB del móvil ocupado.

Problema vigente: error al coincidir el encendido del led con el envío de datos de los sensores

5 Pruebas.

Es necesario un arduino leonardo(o compatible con el protocolo OTG) , el hardware específico para el correcto funcionamiento del código para la estación base y el cable tipo OTG microUSB.

Con todo el hardware correctamente conectado, abriremos la aplicación y pulsaremos en conectarnos, a continuación se nos solicitara el permiso para el uso de dispositivos USB, al aceptar la aplicación mostrará los datos en tiempo real. En caso de desconexión del arduino el aplicación se volverá a conectar automáticamente al conectar el dispositivo nuevamente. Además disponemos de la opción, a modo de configuración de la estación, de encender un led en el propio arduino.

Segunda entrega

1 Funcionalidades.

Las funcionalidades implementadas hasta el momento son las siguientes.

1. Comunicación desde el arduino hacia el dispositivo android.
2. Comunicación desde el dispositivo android hacia el arduino.
3. Refactorización del código de la estación de sensores.
4. Motorización de datos en tiempo real.
5. Modificación de la configuración de la estación.
6. Almacenaje de datos.
8. Adición de nuevos sensores desde el dispositivo android.
9. Guardar configuración de la aplicación.

2 Diseño.

Una única pantalla para visionar los datos en tiempo real, un botón para conectar con el dispositivo y un botón con estado para gestionar una luz en la estación. En caso de perder la conexión con la placa se pide automáticamente la autorización del usuario para reconectar.

Añadido en esta iteración:

Los datos recibidos de los sensores son recolectados en una base de datos. A falta de definir una GUI introducción de nuevos sensores desde el dispositivo android. La aplicación obtiene el estado del led en la estación. La configuración de los sensores es guardada en una base de datos en la aplicación (siendo posible inicializar los sensores con valores por defecto).

3 Planificación.

En esta iteración nos hemos centrado en las bases de datos para la recolección de datos y la configuración de los sensores de la estación (por problemas con la librería gráfica no podemos visualizar el histórico de los datos).

4 Problemas encontrados.

Problemas persistentes desde la iteración anterior:

Se nos ha presentado el problema de que el hardware de la estación de sensores utiliza los pines 10-13 del arduino como salidas SPI, algo que no es compatible con nuestro arduino Leonardo ya que esta versión solo implementa la salida específica de SPI de la placa, lo que nos ha obligado a deshabilitar la pantalla LCD y uno de los sensores. Algo que intentaremos solventar en próximas iteraciones.

Otro de los problemas que nos ha retrasado ha sido el no poder depurar por usb mientras estamos obteniendo datos de la estación, al estar el puerto USB del móvil ocupado (Parcialmente solucionado).

Problemas solucionados:

Problema: error al coincidir el encendido del led con el envío de datos de los sensores. La solución ha sido ignorar la confirmación del envío del mensaje de encendido del led.

Problemas con la depuración: Solventado parcialmente, en cualquier momento podemos desconectar el terminal del arduino y conectarlo al pc, nos dará el log acumulado. En caso de error sin pulsar en aceptar podremos repetir el proceso anterior. En cualquier caso no podremos ver el log en tiempo real.

Nuevos problemas:

Pequeños problemas con las bases de datos, no impiden el correcto funcionamiento del proyecto.

Complejidad con el uso de la librería gráfica: será solventado con en breves.

5 Pruebas.

Es necesario un arduino leonardo(o compatible con el protocolo OTG) , el hardware específico para el correcto funcionamiento del código para la estación base y el cable tipo OTG microUSB.

En caso de querer añadir los sensores por defecto pulsar el botón de inicializar.

Con todo el hardware correctamente conectado, abriremos la aplicación y pulsaremos en conectarnos, a continuación se nos solicitara el permiso para el uso de dispositivos USB, al aceptar la aplicación mostrará los datos en tiempo real. En caso de desconexión del arduino el aplicación se volverá a conectar automáticamente al conectar el dispositivo nuevamente. Además disponemos de la opción, a modo de configuración de la estación, de encender un led en el propio arduino.