FIC – UDC – 1º cuatrimestre 17/18

Proyecto Programación de Sistemas

Easy Arduino!

Jose Luis Álvarez Quiroga – joseluisalvquiroga@gmail.com
Matías Cubilla Currás – matias.cubillac@udc.es

TABLA DE CONTENIDOS

INTRODUCCIÓN

Motivaciones

La motivación principal de este proyecto, es facilitar la utilización e implementación de distintos circuitos Arduino, para cualquier usuario que quiera empezar a utilizarlo, o incluso tenga ya más experiencia en el campo.

“Blynk” es una aplicación que se encarga de esto mismo, a un nivel más avanzado, aunque no sólo trabaja con Arduino, también puede hacerlo con ESP8266, Raspberry Pi, SparkFun, entre otras. Tiene una interfaz bastante trabajada, y también busca simplificar el uso de estos componentes.

Objetivos

El principal objetivo es implementar una versión menos compleja de “Blynk”, enfocado solo a Arduino, facilitando la utilización de los distintos elementos que se puedan utilizar en un Arduino a través de los pines, y todo gestionado a través de una interfaz sencilla y fácil de usar.

El usuario podrá crear distintos paneles, para los distintos circuitos que quiera implementar. Se quedarán guardados en la app, a menos que el usuario decida borrarlos.

Dentro de cada panel, podrá crear, modificar o eliminar los controladores de los distintos elementos y asignarles valores, o en caso de elementos como sensores, podrá gestionar el pedido de información.

El intercambio de información entre el circuito y la app, se llevará a cabo desde el puerto usb del arduino.

Aquí un pequeño ejemplo de lo que se busca.

ANÁLISIS PRELIMINAR DE REQUISITOS

Funcionalidades

Las funcionalidades que se van a implementar, se dividen entre básicas y accesorio, ordenadas en cuánto a la prioridad.

-Básicas:

T1. Creación de la infraestrucutra de comunicación Android-Arduino.

T2. Paso de mensajes a través de la arquitectura anteriormente creada.

T3. Implementación paneles.

T4. Gestión de creación, modificación y eliminación de paneles.

T5. Implementación controladores de envío de datos.

T6. Gestión de creación, modificación y eliminación de controladores de envío de datos.

T7. Almacenamiento en la app de los paneles.

T8. Implementación controladores de lectura de datos.

Dependencias:

T1->T2

T3->T4->T5->T6->T7

-Accesorio:

A1. Cerrar/Abrir conexiones serie con el Arduino al desconectar/conectar USB.

A2. Gestión dinámica de los paneles creados.

A3. Gestión dinámica de los controladores creados.

A4. Retocar interface y acciones para tener un aspecto más fluido.

Dependencias:

T1->A1

T4->A2

T6->A3

Responsabilidades

Jose Luis Alvarez

- Paso de mensajes Android-Arduino
- Interfaz de usuario y UX
- Gestión de paneles y controladores

Matías Cubilla

- Guardar controladores y paneles en memoria del teléfono
- Broadcast Receivers
- Documentación

PLANIFICACIÓN INICIAL

Iteraciones

1ºIteración:

-Implementación de un panel estático inicial..

-Implementación de dos controladores, uno analógico y otro digital.

-Crear infraestructura básica de la comunicación con Arduino, y su Broadcast Receiver.

-Paso de mensajes a través de la infraestructura, sin espera de respuesta.

-Estructura de los mensajes implementada.

-Código .ino que leerá los mensajes.

-Funciones en android que permiten el envío de los mensajes.

Pruebas:

-Establecimiento de conexión correcta.

-Funcionamiento controlador analógico con LED.

-Funcionamiento controlador digital con LED.

Hito: Entrega 09/11/2017.

Entregable:

-Documentación

-Aplicación

2ºIteración:

-Modificar, eliminar y crear paneles.

-Modificar, eliminar y crear controladores.

-Guardado de paneles ya creados con sus controladores

Pruebas:

-Creación, modificación y eliminado de controladores

-Creación, modificación y eliminado de paneles.

-Al cerrar app, se mantienen los paneles

Hito: Entrega 20/12/2017.

Entregable:

-Documentación

-Aplicación

3ºIteración:

-Implementación de controladores de lectura

Pruebas:

-Prueba de funcionamiento de controladores de entrada.

Hito: Entrega final 16/01/2018.

Entregable:

-Documentación

-Aplicación

DISEÑO

Comunicación Android-Arduino

En nuestra aplicación, la comunicación entre Arduino y Android se llevará a cabo a través de USB. Para facilitar la escalabilidad, crearemos un paquete llamado ‘arduinomanager’, donde se incluirá una interfaz, cada una de las implementaciones de dicha interfaz para cada tecnología usada (USB en nuestro caso) y clases adicionales para códigos de respuesta o interfaces que permitan crear listeners de la comunicación.

Comunicación USB

Para poder utilizar la comunicación serie por USB del dispositivo Android usaremos una librería externa, usbserial creada por felHR85 disponible en GitHub.

Arquitecturas utilizadas

La aplicación tendrá implementado un broadcast receiver, que permitirá cerrar la conexión serie con el Arduino cuando se desconecte el cable USB. Así mismo, cuando se conecte un cable USB, se iniciará la conexión.

Para permitir que el layout sea dinámico, haremos uso de fragmentos en distintas partes del proyecto.

Cabe la posibilidad de que se usen threads para ciertas tareas como leer datos del arduino, a pesar de que la librería ‘usbserial’ ya se encarga de gestionar el USB de forma asíncrona.

PRIMERA ITERACIÓN

Arquitectura Propuesta

Comunicaciones:

Comunicación con Arduino utilizando la librería usbserial creada por felHR85 disponible en GitHub.

Vistas:

Sólo una: la del PannelsActivity

Componentes:

-Activities:

-MainActivity

-PannelsActivity

-Broadcast Receiver:

-MyBroadcasteReceiver

-mPermissionReceiver

Diseño e implementación

- Estructura general del proyecto.
- Panel estático y único.
- Dos controladores: uno analógico y uno digital
- Creada arquitectura básica de la comunicación con Arduino
- Paso de mensajes Arduino-Android operativa (sin ningún tipo de respuesta de confirmación todavía)
    - Creada estructura de los mensajes que se mandarán entre ambos (p.ej: *D-W-08-0001)
    - Creado código .ino que leerá dichos mensajes y actuará acordemente
    - Creadas funciones Android que permiten enviar dichos mensajes

Pruebas