

FDS

Índice de contenido

Finalidad del Proyecto.....	3
Requisitos funcionales.....	3
Elementos necesarios.....	3
Extensibilidad del proyecto.....	4
Diseño de la aplicación.....	5
Introducción.....	5
TraductorWIBluePiControl.....	5
Instrucciones de uso.....	5
Comunicaciones.....	8
WIBluePiControl.....	9
Elementos Adicionales.....	10
Librerías usadas.....	10
APENDICES.....	11
REFERENCIAS.....	11

Control puertos GPIO Raspberry Pi

Finalidad del Proyecto

La finalidad principal del proyecto sera tener la capacidad de controlar desde una aplicación movil desarrollada en android los puertos GPIO de un RaspberryPi.

Para esta finalidad se desarrollara los siguientes elementos:

- Un programa que emitirá ordenes en red , ya sea un socket propietario o un por medio de un navegador web a un webservice.
- Un programa en android que recibirá ordenes por red, ya sea un socket propietario o un webservices en android y que se usara de pasarela bluetooth desarrollada en android para comunicar las ordenes a la Raspberry Pi
- Un pequeño servidor de sockects bluetooth desarrollado en Python o Java que interpretara las ordenes recibidas.

Requisitos funcionales

- El sistema debe de ser capaz de establecer la conexión entre los dispositivos android por red LAN, ya sea un socket propietario Cliente/Servidor o por medio de un WebService publicado en android.
- El sistema tiene que se capaz de establecer la conexión entre el dispositivo android y la raspberry por red PAN.
- El sistema tiene que se capaz de activar los puertos GPIO de la RaspberryPi de forma remota.

Elementos necesarios

Para el desarrollo de este proyecto vamos a necesitar los siguientes elementos:

- Raspeberry pi (Ver. B)
- Modulo bluetooth conectado a la Raspberry Pi
- Móvil Android con conexión bluetooth y Wifi.
- Móvil Android con conexión Wifi.
- Leds o motores que consuman menos de 5v para activarlos de forma remota.

Extensibilidad del proyecto

El proyecto puede ser extensible, aunque no se realizara por falta de tiempo, es viable a un siguiente nivel haciendo uso de pequeños componentes electrónicos y realizando un montaje tanto hardware como, electrónico podríamos construir por ejemplo un cuadricoptero, un coche teledirigido

Los elementos necesarios serian estos:

- Una fuente de alimentación de 12V portable
- Unos relés de activación de 5v con capacidad de salida 12v
- Motores de paso
- Display de leds
- Cámara Raspberry
- Placa extensión GPIO Raspberry.

Un proyecto mas funcional y con mas utilidad en el mundo empresarial, podría ser un sistema PTL ((PickToLigth, PutToLigth) económico, lo cual nos daría libertad de movimientos en la preparación de pedidos.

Estos sistemas son usados comúnmente en centros logísticos para la preparación de pedidos.

Para el desarrollo de este sistema tendríamos la necesidad de varios elementos electrónicos :

- Matriz de información, para mostrar datos.
- Pulsadores industriales
- Placa expansión GPIO
- pistola de código de barras USB
- Una fuente de alimentación de 12V portable

Diseño de la aplicación

Introducción

Finalmente se opta por el siguiente esquema de aplicación

Servidor Socket Propietario y cliente Socket Bluetooth propietario (TraductorWIBluePiControl)

Servidor Socket Bluetooth Propietario y Control de GPIO RaspPi programado en Java (WIBluePiControl)

TraductorWIBluePiControl

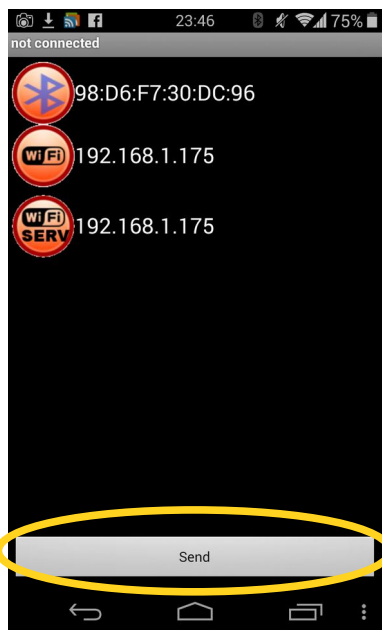
Aplicación Android que traducirá las ordenes recibidas por wifi y las mandara por bluetooth, simplemente es una pasarela y no capacidad de control salvo la de mostrar los mensajes que recibe y envía por las dos partes.

Esta aplicación hace las veces de cliente Wifi conectándose a un socket propietario, es decir podemos hacerla funcionar como cliente de red o como cliente de bluetooth.

Básicamente es un Servidor de socket de red TCP/IP configurado en el puerto 6000 y dos clientes de socket uno bluetooth y otro TCPIP.

Instrucciones de uso

La aplicación se compone de las siguientes pantallas



Pantalla principal de la aplicación.

En esta pantalla se observan las distintas conexiones disponibles así como las direcciones locales de las conexiones.

Estos elementos activos y permiten realizar las conexiones



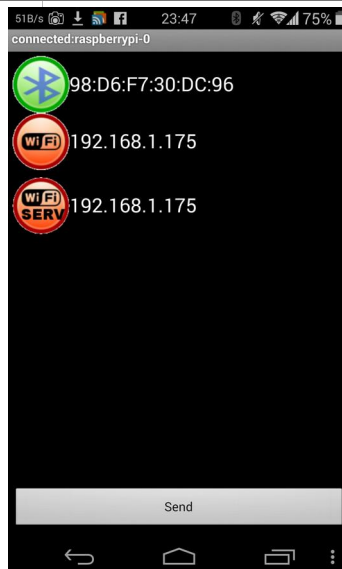
Se aprecia el menú de conexión de Bluetooth desde el botón de menú del móvil. Se aprecia que tenemos un elemento en el menú del móvil para indicarnos que estamos conectados, remarcado de color verde.



Al pulsar el elemento de menú Conectar Bluetooth, se muestra la pantalla anterior, se mostrara el menú de conexión de Bluetooth remarcadas de color verde. Al pulsar el elemento de menú Conectar Bluetooth, se mostrara el menú de conexión de Bluetooth remarcadas de color verde. Al pulsar el elemento de menú Conectar Bluetooth, se mostrara el menú de conexión de Bluetooth remarcadas de color verde.



lanzadores marcados en naranja
diálogo de conexión en red donde
ir la ip de dispositivo que hace de
to 6000.



los ver el detalle de un dispositivo
la aplicación. Los demás iconos
se este o no conectado.



Dispositivo conectado al servidor de socket



Dispositivo conectado a Bluetooth



Dispositivo conectado como cliente de red



Comunicaciones

Al ser un componente dual se han unificado todos los mensajes que se pueden mandar desde los dos clientes.

Identificamos tres posibles orígenes y destinos de la aplicación:

RPI	Identificador con el que definimos que el origen o el destino es el dispositivo RaspberryPi.
MOVIL	Identificador con el que definimos que sera el dispositivo que recibirá los SMS
TRADUCTOR	Dispositivo que hace de puente entre Bluetooth y red lan

Tenemos los siguientes mensajes disponibles

Mensaje	Destino	Parámetros	Descripción
ON	RPI	Id de 1 a 4 identificando el led a encender	Enciende el led indicado
OFF	RPI	Id de 1 a 4 identificando el led a encender	Apaga el led indicado
HELLO	RPI	NONE	Activa una secuencia de saludo con los leds
SET	TRADUCTOR RPI	Numero de móvil	Marca el móvil de envío de sms
SMS	MOVIL RPI	Móvil , mensaje	Manda el mensaje indicado al móvil introducido. En caso de que el destino se RPI esta reenviara el telegrama al móvil indicado. Este mensaje es también activado cuando en la raspberry pulsamos el botón que tiene conectado al puerto GPIO 0
SOUND	No implementado	No implementado	No implementado
ALLON	RPI	NONE	Enciende todas las luces
ALLOFF	RPI		Apaga todas las luces
QUIT	No implementado	No implementado	No implementado

WIBluePIControl

Aplicación Java que esta a la espera de ordenes tanto por BlueTooth como por WIFI y realizara las acciones que se le manden por los sockets, activando o desactivando los puertos GPIO, asimismo estará escaneando si alguno de los puertos configurados es activado desde la raspberry para mandar una alarma de control a los sistemas superiores.

Elementos Adicionales

Para poder usar las librerías de bluetooth en Raspberry PI hemos tenido que compilar la librerías de sistema BlueCove para plataforma ARM (<http://bluecove.org/bluecove-gpl/index.html>).

Dado que bajo interfaz linux ARM el sistema de bluetooth no pide autenticación se han tenido que vincular los dispositivos bluetooth a mano por linea de comandos, se ha seguido el siguiente manual (<http://www.diverteka.com/?p=1880>) los comandos usados han sido los siguientes:

hcitool dev	Nos muestra la direccion local
hcitool scan	Buscamos dispositivos bluetooth
sudo bluez-simple-agent hci0 11:22:33:44:55:66	Conectamos a la mac encontrada
sudo bluez-simple-agent hci0 11:22:33:44:55:66 repair	Reconectamos

Tambien se han conectado una serie de leds al puerto GPIO de una raspberry pi y un botón pulsador que nos hará las veces de alarma.

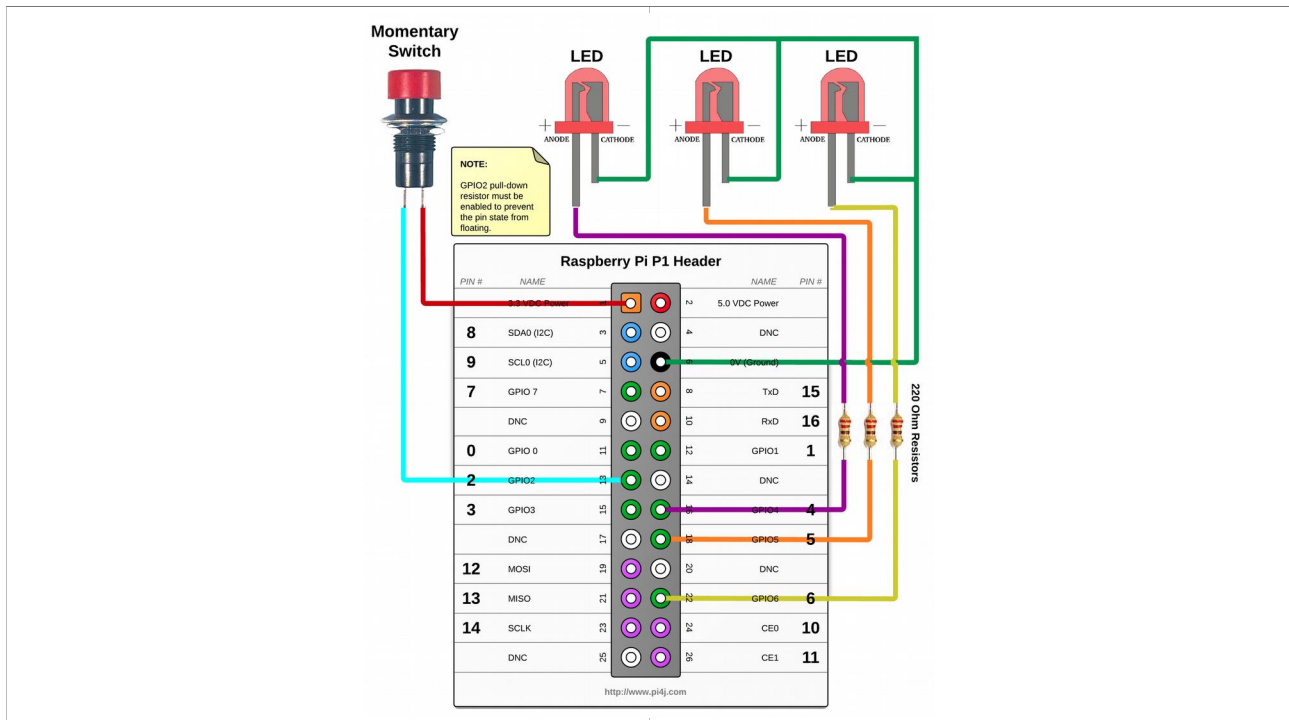
Librerias usadas

Se han usado para la aplicación las siguientes librerías :

- PI4J Libreria usada para conectar y manejar los puertos GPIO de Raspberry Pi desde una aplicación java. “<http://code.google.com/p/pi4j/downloads>”
- BlueCove : Librería que permite controlar un dispositivo bluetooth desde java, dado que no existe un driver estandar compilado para plataforma ARM, se ha tenido que compilar la librería en el dispositivo Raspberry para generar un driver capaz de unir las dos partes , sistema y programación. “<http://bluecove.org/>”
- Log4J en la parte de Raspeberry se ha usado esta librería para permitir ver los evento que suceden durante la ejecucion.

APENDICES

Imagen de los puertos IO de la RaspberryPI y esquema de la instalación.



REFERENCIAS.

http://code.google.com/p/pi4j/downloads	Documentación y ejemplo sobre el acceso a GPIO desde Java
http://pi4j.com/	Documentación oficial sobre GPIO en Java
http://bluecove.org/	Documentación oficial de librerías de bluetooth en java
http://www.diverteka.com/?p=1880	Manejo de bluetooth desde linea de comandos
http://androideity.com/	Documentación y ejemplo de programación android
http://stackoverflow.com	Base de datos de expertos en programación resuelve muchas dudas sobre situaciones extrañas