

SISTEMAS OPERATIVOS AVANZADOS



Universidad Nacional de La Matanza

SISTEMA DE RIEGO INTELIGENTE

INTEGRANTES:

Delfino Julian

Garcia Pena Facundo

Nicoloso Lucas

Martinez Lucas

Índice

1. Identificación del trabajo	3
2. Introducción	3
Necesidad.....	3
Objetivo.....	3
Alcance	3
Modo Automático	4
Modo Manual.....	4
Límite	4
Modo Automático	4
Modo Manual.....	4
Software utilizado	5
Para el Sistema Embebido:	5
Para Android:	5
Hardware utilizado.....	5
Diagramas	6
Diagrama Funcional	6
Diagrama Software	7
Esquema de conexión	8
Sensores	9
Sensor de Humedad FC-28.....	9
Sensor de lluvia Raindrop	10
Fotoresistor LDR.....	11
Android	12
Interfaces	12
Detalle Técnico de la Aplicación	13

1. Identificación del trabajo

Desarrollo de un Sistema de Riego Inteligente que le permita al usuario la ejecución manual o automática de riego en tres circuitos diferentes pudiendo manipular lo mencionado desde un Dispositivo con Sistema Operativo Android o con el teclado y pantalla que el Sistema ofrece.

2. Introducción

Necesidad

Gracias al avance de la tecnología, hoy en día se busca automatizar tareas cotidianas con el fin de que se puedan lograr de una manera más simple y eficaz. Por esta razón se comenzó con el Proyecto de Sistema de Riego Inteligente logrando abstraer al Usuario de esta tarea rutinaria ya sea para su hogar o para un espacio mediano teniendo la posibilidad de interactuar con el Sistema mediante un Dispositivo Android o desde un teclado y pantalla que el Sistema ofrece.

Objetivo

Cada circuito posee un sensor de humedad el que nos brinda la humedad de la tierra, un sensor de luz el cual nos brinda detalladamente la intensidad lumínica que hay en un determinado momento y un sensor de lluvia que se encarga de detectar el volumen de agua precipitada. Por otro lado, cada circuito posee un relé como actuador el cual permitirá el paso del agua en el momento que sea necesario. Por último, el Usuario puede establecer excepciones de ejecución según los valores actuales de los Sensores.

El Usuario tiene la posibilidad de interactuar con el Sistema mediante una Aplicación Android o desde un teclado y pantalla que el Sistema ofrece.

El Sistema estará controlado por un Arduino Mega, el cual se encargará de obtener los datos de los sensores mencionados, procesarlos y ejecutar los actuadores en caso que corresponda según su configuración y excepciones.

Alcance

Dividiremos el Alcance según el modo de ejecución:

Modo Automático

Para que el modo automático entre en funcionamiento, el Usuario debe establecer los límites que desee en los Sensores como así también las excepciones de días y horario. Partiendo de esta configuración, el Sistema de Riego Inteligente en modo automático comenzará a regar. Su ejecución terminará si:

- El sensor de Humedad supera el límite fijado o
- El sensor de precipitación supera el límite fijado o
- La luz del ambiente (LDR) supera el límite fijado.

Modo Manual

Para que el modo manual entre en funcionamiento, el Usuario debe ejecutarlo ya sea desde la Aplicación de Android o desde el teclado que el Sistema Ofrece. Las excepciones no se consideran en esta modalidad.

Límite

Modo Automático

Desde que el Usuario establece los límites de ejecución y excepciones en el Sistema

Hasta que finaliza la ejecución del Sistema automáticamente cuando llega a los valores máximos establecidos.

Modo Manual

Desde que el Usuario pone en ejecución el Sistema de Riego Inteligente

Hasta que finaliza la ejecución del Sistema manualmente.

Software utilizado

Para el Sistema Embebido:

Microsoft Windows 10

Arduino Genuino 1.8.5

Para Android:

IDE Android Studio

Hardware utilizado

Placa Arduino MEGA

3 Sensor de Humedad del suelo FC-28

Sensor de lluvia Raindrop

Fotoresistor LDR

3 Led Rojo

1 Led Azul (PWM)

RTC DS3231

LiquidCrystal I2C

Teclado 4x4

5 resistencias 220 Ohm

Cables Macho – Macho

Cables Macho – Hembra

Protoboard pcb

Bluetooth HC-05

3 Módulos de Relés

Diagramas

En esta sesión se describirán los diagramas utilizados para comenzar a desarrollar el proyecto.

Diagrama Funcional

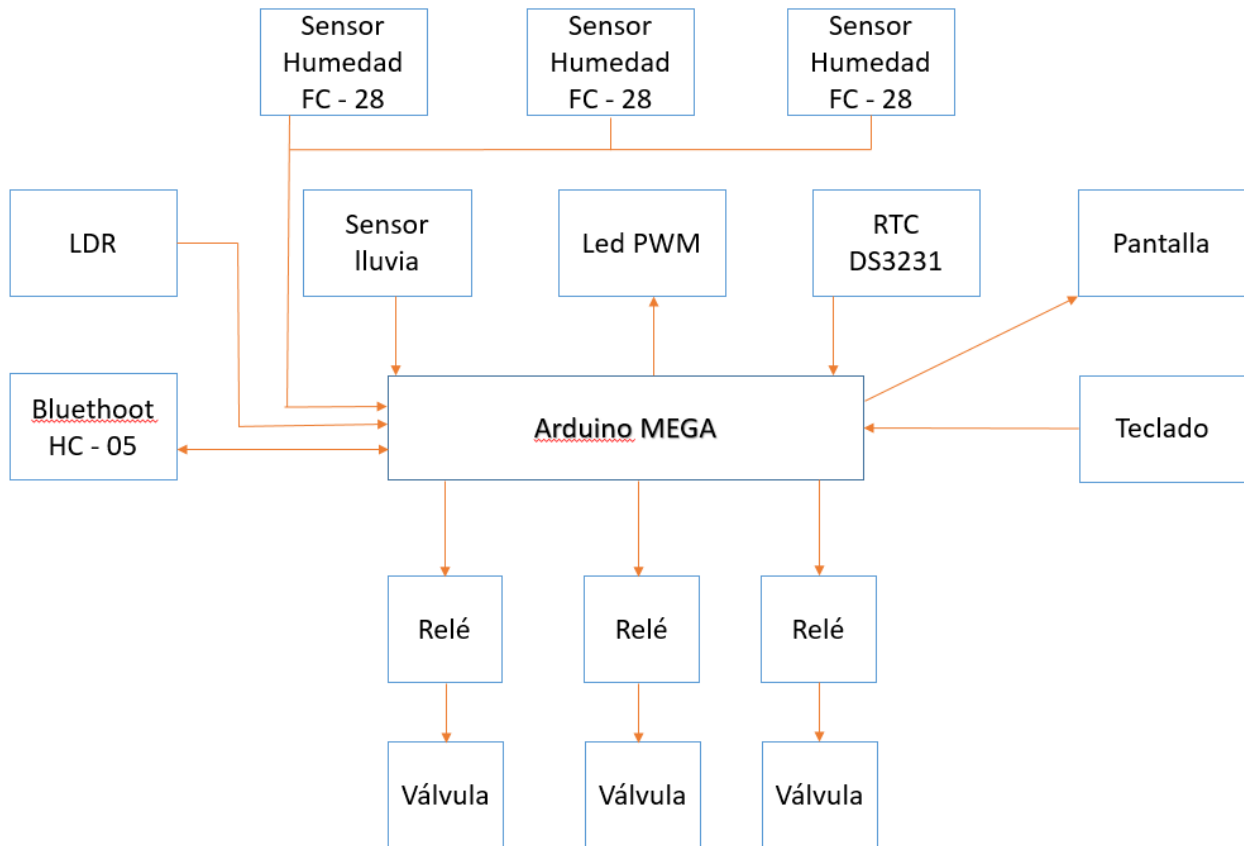
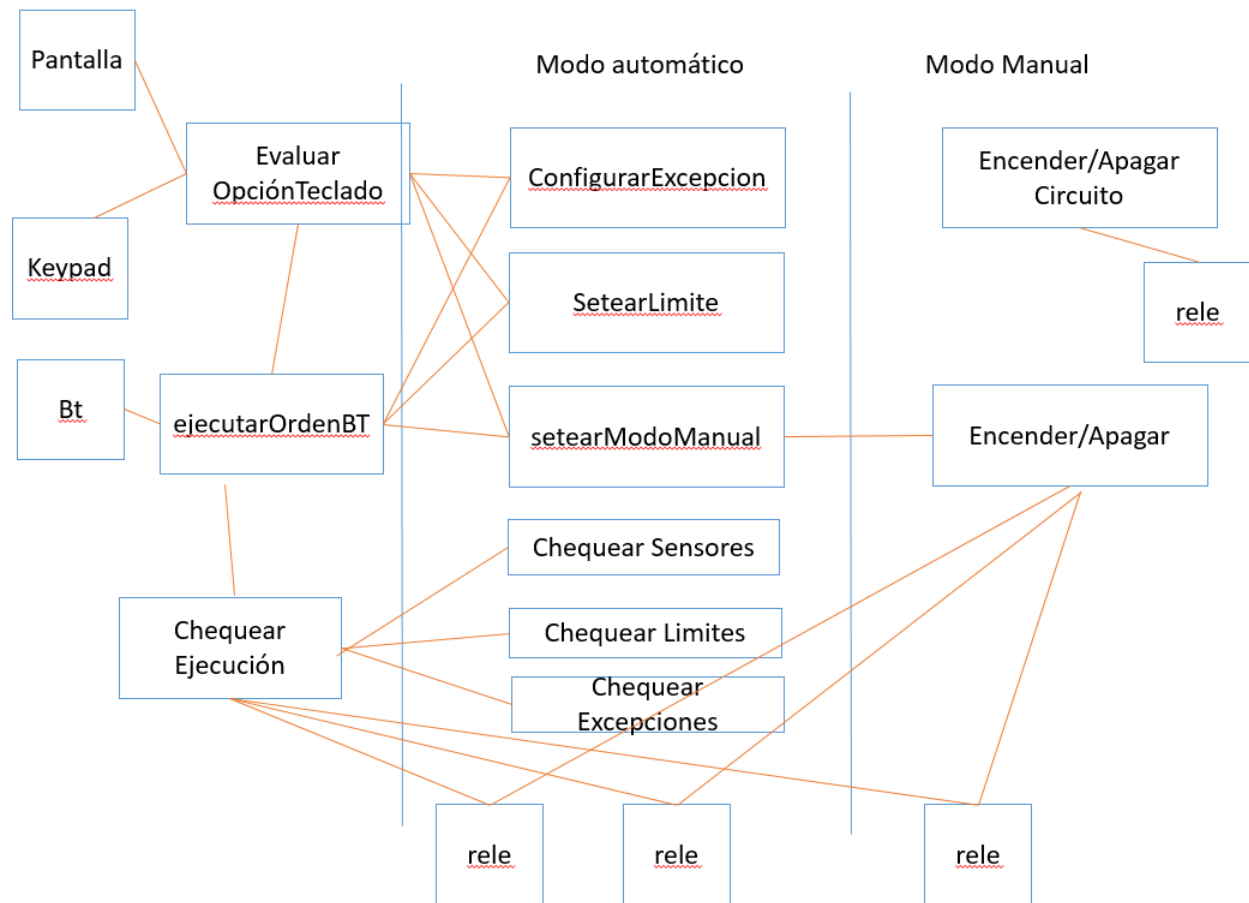
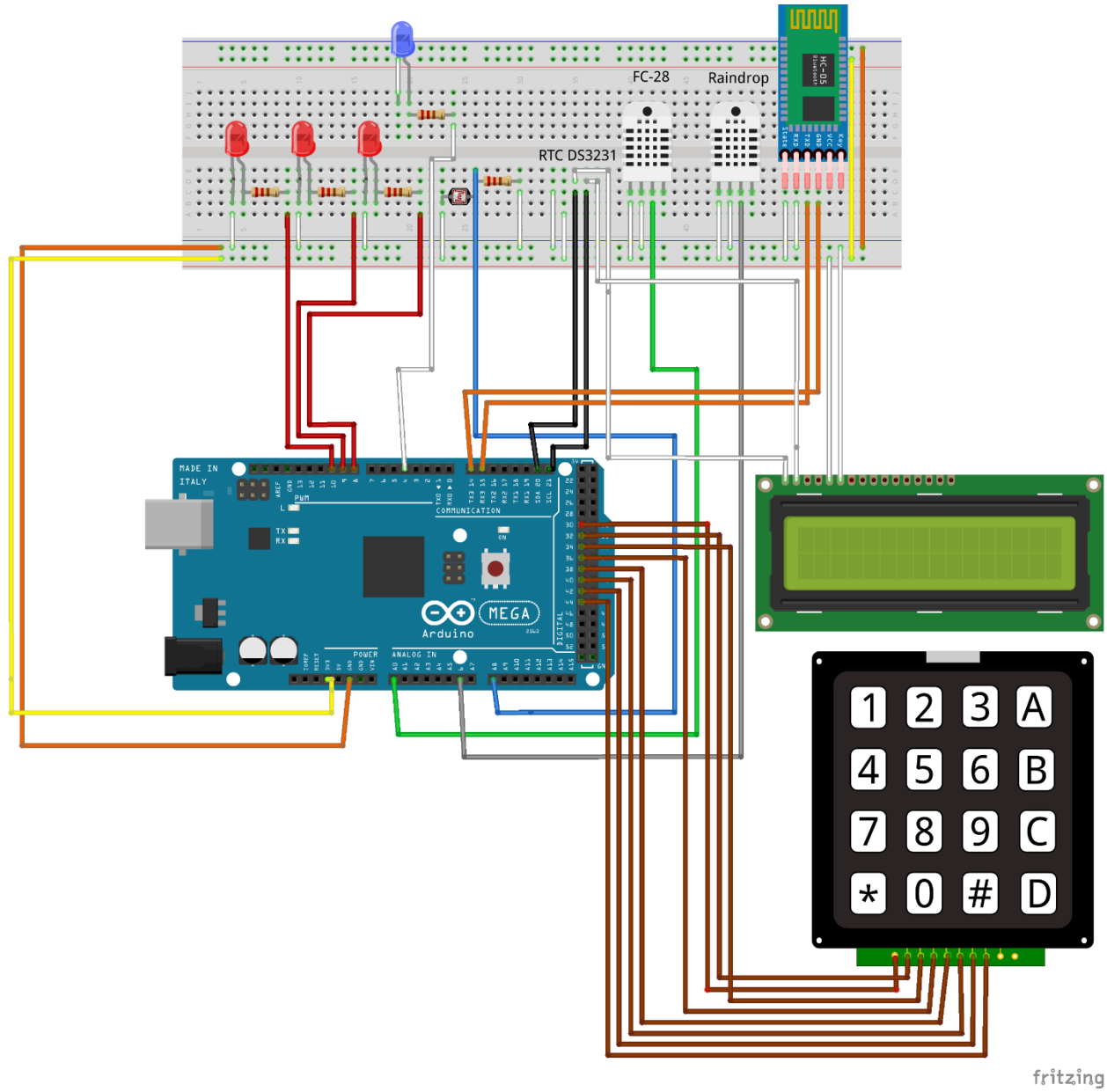


Diagrama Software



Esquema de conexión



Sensores

En esta sesión se hará una breve descripción de cada sensor que interactúa en nuestro sistema.

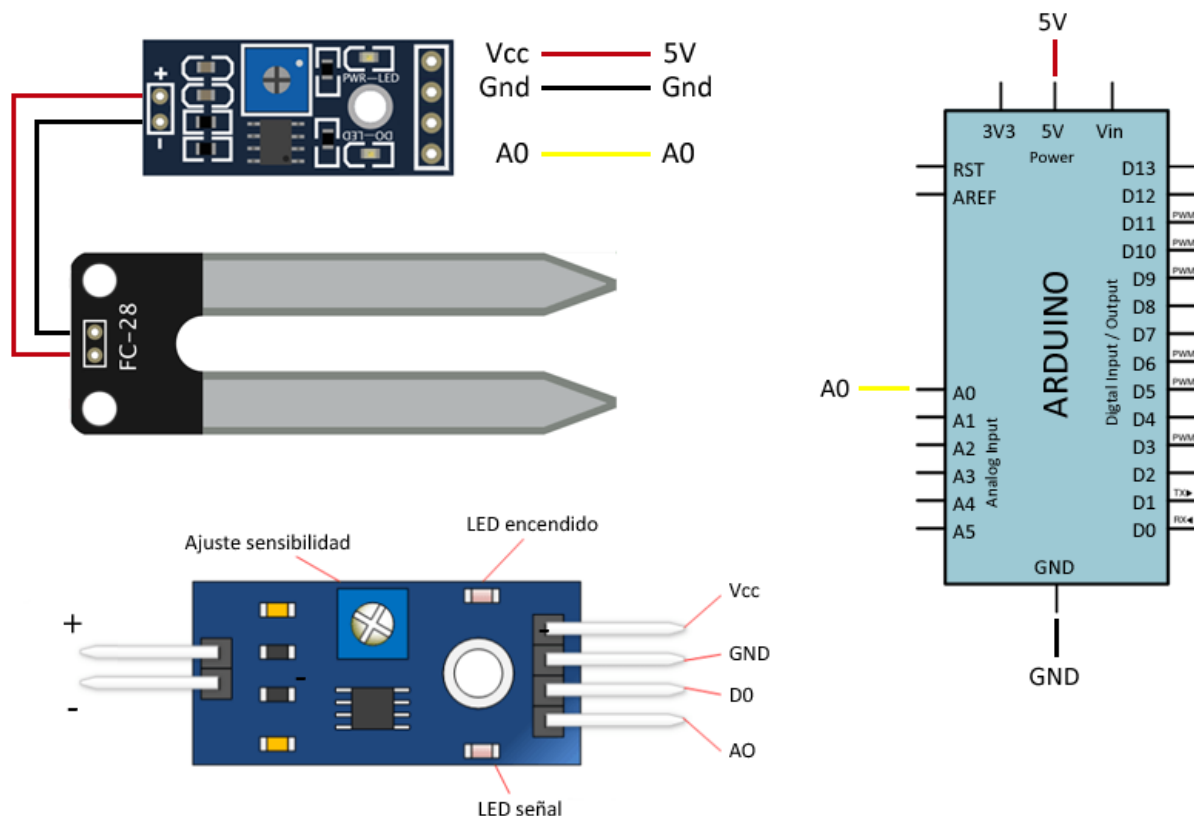
Sensor de Humedad FC-28

Encargado de medir la humedad del suelo por la variación de su conductividad. Particularmente este sensor no tiene la precisión suficiente para realizar una medición absoluta de la humedad del suelo pero tampoco es necesario para controlar nuestro sistema de riego.

Su salida puede ser analógica (Valores desde 0 a 1023) o digital.

Conexión a Arduino

Si la conexión será analógica debemos conectar la salida A0 y si es digital conectamos a D0 y con el potenciómetro de la placa ajustamos la sensibilidad.



Sensor de lluvia Raindrop

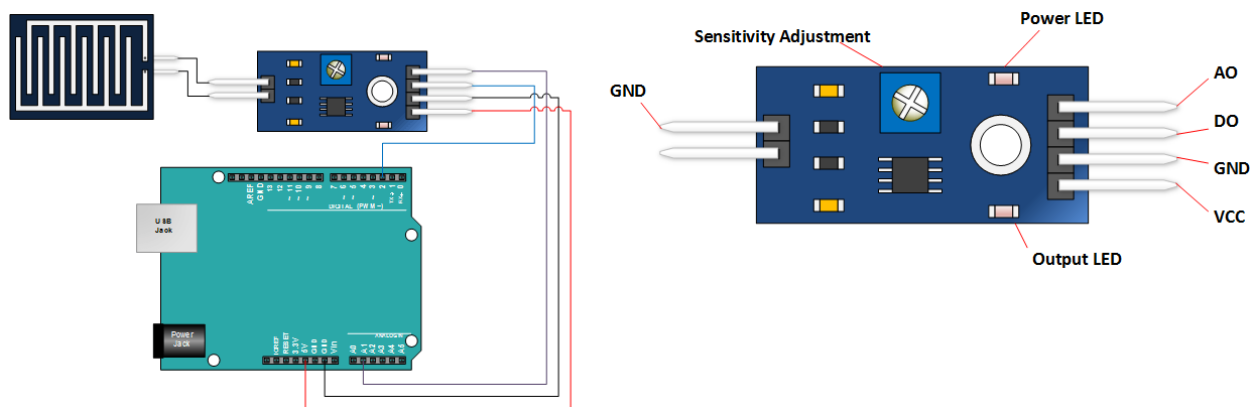
Detectan la presencia de lluvia por la variación de conductividad del sensor al entrar en contacto con el agua.

Constructivamente son sensores sencillos, se dispone de dos contactos unidos a unas pistas conductoras entrelazadas entre sí a una pequeña distancia, sin existir contacto entre ambas. Al depositarse agua sobre la superficie, se pone en contacto eléctrico ambos conductores, lo que puede ser detectado por un sensor.

Su salida puede ser tanto analógica (1023 no hay lluvia, 0 lluvia) como digital.

Conexión a Arduino

Si la conexión será analógica debemos conectar la salida A0 y si es digital conectamos a D0 y con el potenciómetro de la placa ajustamos la sensibilidad.



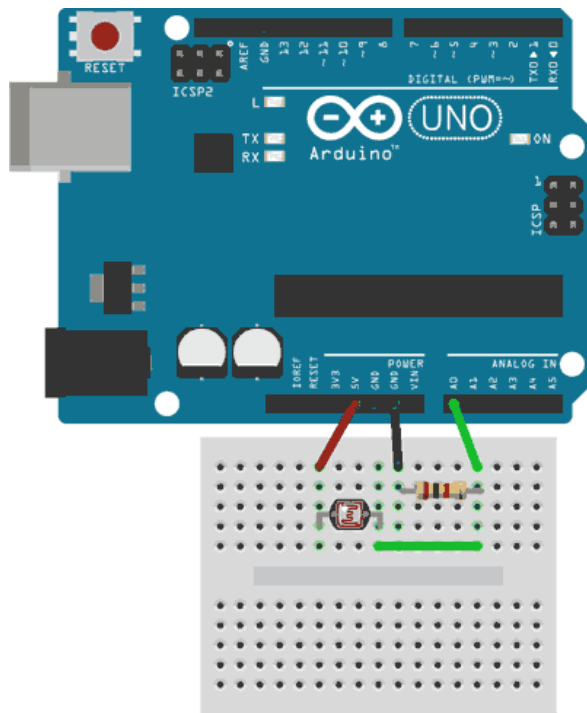
Fotoresistor LDR

Es un dispositivo cuya resistencia varía en función de la luz recibida. Podemos usar esta variación para medir, a través de las entradas analógicas, una estimación del nivel de luz.

Está formado por un semiconductor, típicamente sulfuro de cadmio. Al incidir la luz sobre él, algunos de los fotones son absorbidos, provocando que electrones pasen a la banda de conducción y, por lo tanto, disminuyen la resistencia del componente. Por lo tanto, disminuye su resistencia a medida que aumenta la luz sobre él.

Sus valores típicos son de 1 Mohm en total oscuridad, a 50-100 Ohm bajo luz brillante.

Su variación de resistencia es relativamente lenta haciendo así que no sea posible registrar variaciones rápidas, como las producidas en fuentes de luz artificiales alimentadas por corriente alterna.

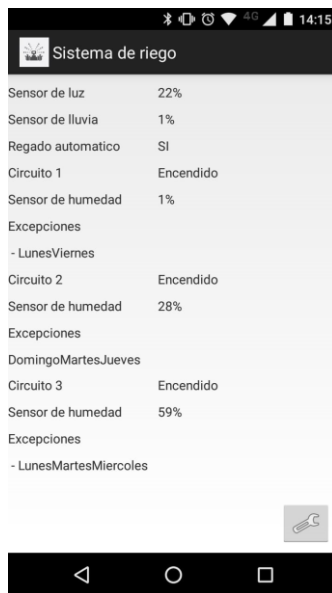


Android

Para manipular el Sistema de Riego, se desarrolló una aplicación en Android. A continuación, se comparten algunas capturas mostrando el funcionamiento.

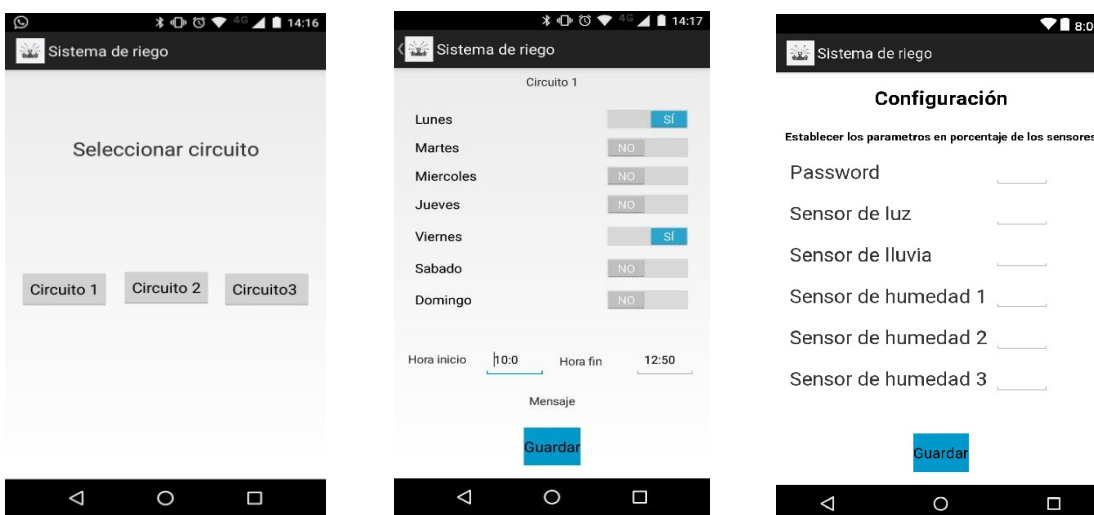
Interfaces

Visualización del estado de los Circuitos en tiempo real



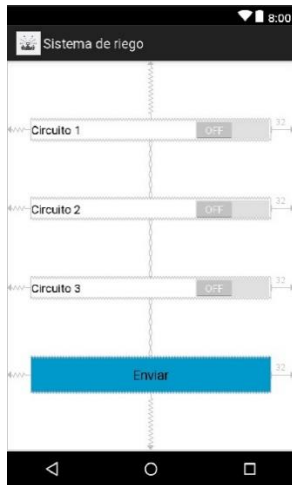
Como se puede observar en la imagen, la aplicación es capaz de mostrar en tiempo real el estado actual de sus tres circuitos con sus respectivos sensor de luz, lluvia, humedad y las excepciones establecidas en cada uno de ellos. A esta portada llegamos cuando ya tenemos conexión con el Bluetooth.

Configuración de parámetros y excepciones de los Circuitos para el modo automático



Como se puede observar en la imagen, la aplicación permite configurar las excepciones de ejecución de cada circuito cuando se ejecuta este en modo automático.

Activación de circuitos manualmente



Como podrán observar en la imagen, además del modo automático ya mencionado, se pueden activar los circuitos de manera manual.

Detalle Técnico de la Aplicación

Conectividad

Nuestro sistema embebido funcionará como Cliente de nuestra aplicación Android. Esta última se encargará de solicitar toda la información que sea necesaria y le indicará que es lo que tiene que hacer. La comunicación entre ellos será mediante Bluetooth.

Entando en detalle a la comunicación, ante la necesidad de en distintos activities utilizamos un `IntentService`, que es un servicio que atiende pedidos por medio de una cola. En este servicio está toda la lógica de bluetooth, y se comunica con las otras activities utilizando un `BroadcastReceiver`, que debe registrarse en cada activity.

El método `onHandleIntent` que se usa desde el `IntentService`, sirve para mandarle mensajes por el broadcast a la activity, especificando el tipo de broadcast al cual debe enviar.

Protocolo de comunicación

Se estableció un “lenguaje” de comunicación entre el Arduino y la aplicación de Android para saber el estado actual del Sistema, obtener históricos, excepciones, ejecuciones en modo automático o manual, etc. A continuación, se adjuntan los mensajes correspondientes al protocolo de comunicación establecido:

La aplicación en Android solicita el estado general de los sensores, funcionamiento, excepciones y modo de ejecución:

```

ANDROID --> CMD|STATUS#
EMBEBIDO --> CMD|STATUS|luz|lluvia|SI/NO(modo automatico activado)#
EMBEBIDO --> CMD|1(numero de circuito)|humedad|SI/NO (si esta funcionando el
             circuito)|dias(0,1,2,3,4,5,6)|excepcion hora desde|excepcion minutos desde|excepcion hora
             hasta|excepcion minutos hasta#
EMBEBIDO --> CMD|2(numero de circuito)|humedad|SI/NO (si esta funcionando el
             circuito)|dias(0,1,2,3,4,5,6)|excepcion hora desde|excepcion minutos desde|excepcion hora
             hasta|excepcion minutos hasta#
EMBEBIDO --> CMD|3(numero de circuito)|humedad|SI/NO (si esta funcionando el
             circuito)|dias(0,1,2,3,4,5,6)|excepcion hora desde|excepcion minutos desde|excepcion hora
             hasta|excepcion minutos hasta#

```

La aplicación en Android solicita los últimos 10 valores históricos sobre la ejecución:

```

ANDROID --> CMD|HISTORIA#
EMBEBIDO --> CMD|HISTORIA|1 a 10 (nº de muestreo)|luz|lluvia|humedad 1|humedad 2|humedad 3|SI/NO
             (excepcion 1 activada)|SI/NO (excepcion 2 activada)|SI/NO (excepcion 3 activada)|SI/NO (si
             esta funcionando el circuito 1)|SI/NO (si esta funcionando el circuito 2)|SI/NO (si esta
             funcionando el circuito 3)|hora de lectura(formato HH:MM:SS)#

```

La aplicación en Android solicita el estado de las configuraciones de los limites de sensores, código de ingreso y excepciones de cada circuito:

```

ANDROID --> CMD|CONFIG#
EMBEBIDO --> CMD|CONFIG|limite sensor LDR (luz)|limite sensor lluvia|codigo de ingreso(XXXX)#
EMBEBIDO --> CMD|1|limite sensor humedad 1|dias(0,1,2,3,4,5,6)|excepcion hora desde|excepcion minuto
             desde|excepcion hora hasta|excepcion minutos hasta#
EMBEBIDO --> CMD|2|limite sensor humedad 2|dias(0,1,2,3,4,5,6)|excepcion hora desde|excepcion minuto
             desde|excepcion hora hasta|excepcion minutos hasta#
EMBEBIDO --> CMD|3|limite sensor humedad 3|dias(0,1,2,3,4,5,6)|excepcion hora desde|excepcion minuto
             desde|excepcion hora hasta|excepcion minutos hasta#

```

La aplicación en Android solicita el estado de las excepciones (días y horas) de los diferentes circuitos:

```

ANDROID --> CMD|EXC#
EMBEBIDO --> CMD|EXC|1|dias(0,1,2,3,4,5,6)|hora desde|minutos desde|hora hasta|minutos hasta#
EMBEBIDO --> CMD|EXC|2|dias(0,1,2,3,4,5,6)|hora desde|minutos desde|hora hasta|minutos hasta#
EMBEBIDO --> CMD|EXC|3|dias(0,1,2,3,4,5,6)|hora desde|minutos desde|hora hasta|minutos hasta#

```

La aplicación en Android setea los valores de pin de ingreso (1), los límites de los sensores (2) y las excepciones de algún circuito en particular (3):

```

(1)ANDROID --> CMD|SET|PIN|XXXX#
(2)ANDROID --> CMD|SET|LIMITE|limiteLuz|limiteLluvia|limite humedad 1|limite humedad 2|limite humedad 3#
(3)ANDROID --> CMD|SET|EXC|numero circuito|0,1,2,3,4,5,6 (días)|hora desde|minuto desde|hora
             hasta|minuto hasta#

```

La aplicación en Android Enciende/Apaga el modo automático de ejecución del sistema:

```
ANDROID --> CMD|AUTO|ON#  
ANDROID --> CMD|AUTO|OFF#
```

Si el modo manual está activado, la aplicación en Android Inicia/Para la ejecución de un circuito manualmente:

```
ANDROID --> CMD|INICIAR|numero de circuito#  
ANDROID --> CMD|PARAR|numero de circuito#
```

Activities principales

- MainActivity: Activity principal de la aplicación.
- DeviceListActivity: Encargada de mostrar los dispositivos Bluetooth encontrados.
- BluetoothService: Dispone toda la configuración del Servicio Bluetooth y los hilos.

Sensores utilizados

- Proximidad
- Shake (Acelerómetro)
- Luminosidad

Hilos utilizados

1. Hilo que maneja la aplicación principal.
2. Hilo que maneja el intent service