

Automatización, monitoreo y control remoto de un sistema de riego agrícola con código abierto

Bladimir J. Pérez

Universidad de Panamá, Panamá, Panamá, bladimir.perezq@gmail.com

José M. Koo

Universidad de Panamá, Panamá, Panamá, jose.koo@ieee.org

Fernando García

Universidad Carlos III de Madrid, Leganés, Madrid, España, nandogf@gmail.com

Juan Carmona

Universidad Carlos III de Madrid, Leganés, Madrid, España, jucarmon@ing.uc3m.es

ABSTRACT

The research objective is to obtain a prototype of low cost of a system agricultural irrigation automated, monitored and controlled remotely from anywhere in the world where there is cell phone service. The main tools used are: Arduino and Android. Arduino is responsible for system automation and Android facilitate user interaction showing system data, giving you the option to remotely monitor and control the irrigation system from the Internet, anywhere in the world, achieving security and flexibility to the farmer.

Keywords: Arduino, Android, ZigBee, XBee, Soil Moisture Sensor, open source, automatization.

RESUMEN

El objetivo de la investigación, es obtener un prototipo de bajo costo de un sistema de riego agrícola automatizado, monitoreado y controlado remotamente desde cualquier parte del mundo donde exista servicio de telefonía celular. Las principales herramientas utilizadas son: Arduino y Android. Arduino se encargará de la automatización del sistema y Android facilitará la interacción con el usuario, mostrándole los datos del sistema, brindándole la opción de poder monitorear y controlar el sistema de riego a través de internet desde cualquier parte del mundo, logrando seguridad y flexibilidad al agricultor.

Palabras claves: Arduino, Android, ZigBee, XBee, Soil Moisture Sensor, código abierto, automatización.

1. INTRODUCCIÓN

En la actualidad los sistemas de riego, en su gran mayoría, son operados de forma manual, requiriendo mucha atención del agricultor. Básicamente el agricultor por su experiencia decide el periodo de tiempo y el momento de regar el cultivo. Debido a esto se pueden presentar problemas comunes generados por el ser humano, como por ejemplo, olvidar encender y apagar el sistema en el tiempo considerado. Lo anterior causa problemas directos con el cultivo, como por ejemplo, exceso de agua al cultivo, lo que acarrearía mayor consumo del recurso hídrico que es tan importante a nivel nacional e internacional o bien secándolo por falta de la misma, además de un mayor consumo energético del sistema. Todo lo anterior disminuye la productividad, causándole pérdidas económicas al agricultor. Teniendo un sistema automatizado, se busca solucionar estos errores, aumentando la producción, ahorrar en el consumo energético y minimizar la merma en la productividad, ya que el agricultor no tendrá que estar pendiente permanentemente del cultivo, porque el sistema al ser autónomo tomaría las decisiones requeridas.

Además si al desarrollo se le añade la opción de monitoreo y control remoto a través de una aplicación móvil, el agricultor se sentirá cómodo y seguro de que su cultivo está siendo regado adecuadamente.

2. OBJETIVOS

- Desarrollar un prototipo de un sistema de riego agrícola automatizado, capaz de ser monitoreado y controlado remotamente desde una aplicación móvil.
- Elegir el microcontrolador, sensores, tecnología inalámbrica y demás herramientas que nos ayuden a desarrollar la investigación.
- Utilizar herramientas de bajo costo y alta flexibilidad.
- Programar en lenguajes de acceso libre, alta disponibilidad y escalabilidad.

3. MÉTODOS Y MATERIALES

3.1 ARDUINO MEGA ADK (ANDROID DEVELOPERS KIT)

Basado en el Arduino Mega 2560, con la diferencia de que sus desarrolladores lo crearon pensando en trabajar directamente con dispositivos Android, de forma práctica, sencilla y flexible.

Esta placa a diferencia de los demás Arduinos, cuenta con un puerto USB adicional, que funciona como host USB para comunicarse con dispositivos Android. La coordinación del enlace se logra a través del chip MAX3421e incorporado en la placa ADK y utilizando librerías proporcionadas tanto por Arduino como por Android, razón principal por la cual se eligió este hardware (Böhmer, 2012).

Como nodo principal de nuestro sistema recibirá los datos de los diferentes sensores, permitiéndonos de esta manera la automatización del mismo, además de transferir los datos al dispositivo Android.

Tendrá conectado los siguientes dispositivos: módulo XBee, tarjeta de relay, sensor de flujo, buzzer, pantalla LCD 16x2, los cuales se explicaran más adelante.

El Arduino Mega ADK luce como en la Figura 1.

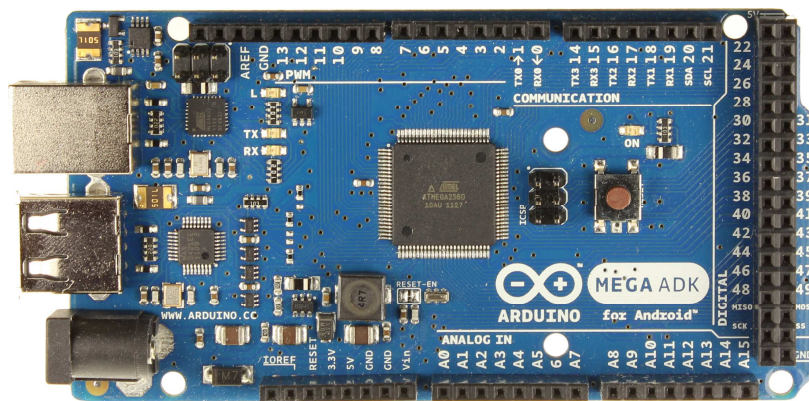


Figura 1: Arduino Mega ADK

3.2 ARDUINO UNO

Arduino Uno es una de las plataformas más utilizadas y la adecuada para comenzar con todo lo relacionado a esta tecnología (Torres, 2013). Ha recibido varias actualizaciones y la más reciente es la revisión número tres (R3), que es la que utilizaremos en esta investigación.

Decidimos elegir este modelo como nodo secundario de nuestro sistema y tendrá conectados: un módulo XBee, Shield Wireless y el sensor de humedad de suelo.

Se encargará de capturar el valor de humedad, a través del sensor y enviará este dato por medio del módulo XBee, al Arduino Mega ADK.

3.3 XBEE

Modulo basado en la tecnología inalámbrica ZigBee (Valverde, 2007). Utiliza el protocolo IEEE 802.15.4 para comunicaciones punto a punto y punto a multipunto. También cuenta con una baja latencia de transmisión, bajo consumo energético y largo alcance. Además, se pueden configurar para implementar una red Mesh para futuros desarrollos (Montesinos, 2013).

Construidos y respaldados por la empresa Digi's International, que cuenta con toda la documentación en línea y también con el software gratuito compatible con Windows, para la configuración de sus parámetros.

Existen dos tipos de módulos XBee, denominados: serie 1 y serie 2. Los mismos no son compatibles entre sí y no pueden comunicarse entre sí. En este trabajo utilizaremos la serie 2 (modo API), ya que es la que cuenta con menor consumo energético y mayores opciones avanzadas.

Los nodos XBee se pueden configurar de dos modos denominados: modo AT y API.

- Modo AT (Attention). En esta configuración los datos que se desean transmitir no reciben ningún tipo de modificación. El paquete puede ser enviado a un solo destino o a múltiples destinos.
- Modo API (Application Programming Interface). Este modo realiza un encapsulamiento de los datos, proporcionando opciones adicionales como: checksum, direccionamiento y confirmación del paquete transmitido/recibido.

Su papel en nuestro esquema será gestionar la transmisión y recepción de datos de humedad del suelo entre nuestros Arduinos.

A continuación muestro en la Figura 2 la apariencia del módulo comentado.

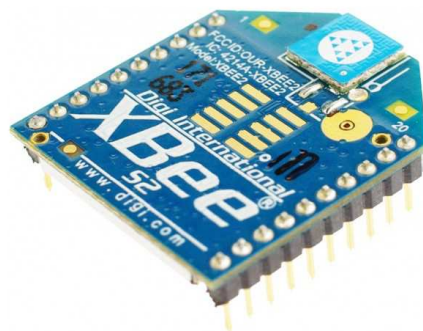


Figura 2: XBee Serie 2

3.4 SENSOR DE FLUJO

Deberá estar instalado en la tubería de agua principal del sistema de riego, monitoreando el flujo de agua y de esta forma vigilar el sistema de bombeo.

3.5 SOIL MOISTURE SENSOR

Es un tipo de sensor resistivo, que utiliza dos sondas para medir la corriente que circula en el medio, a través del cual podemos determinar la resistencia, con lo cual se calcula la humedad del terreno. El porcentaje de humedad será mayor mientras más cantidad de agua esté presente en la tierra (Glaría and Kouro, 2001).

Este sensor estará conectado a cada Arduino Uno que se establezca como nodo secundario. Los valores capturados son de tipo análogo, enviando un valor entero de 0 a 950. Si el suelo está seco, obtendremos un valor entero de 0 a 300, si está húmedo de 301 a 600 y si está saturado de 601 a 950.

A continuación en la Figura 3 muestro la apariencia del sensor utilizado.

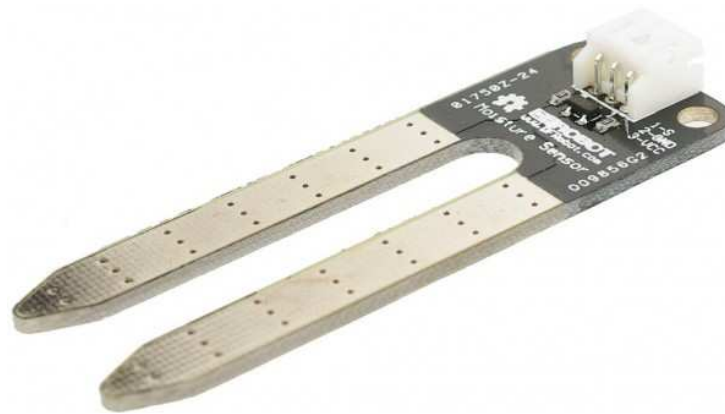


Figura 3: Soil Moisture Sensor

3.6 WIRELESS SHIELD

Utilizado para mayor comodidad de conexión entre el Arduino y el módulo XBee. También es necesario para configurar nuestros módulos XBee, ya que contiene un interruptor que consta de dos modos: micro y USB. Para configurar el módulo XBee a través del software del fabricante es necesario colocar este interruptor en modo USB.

La Figura 4 representa el adaptador comentado.

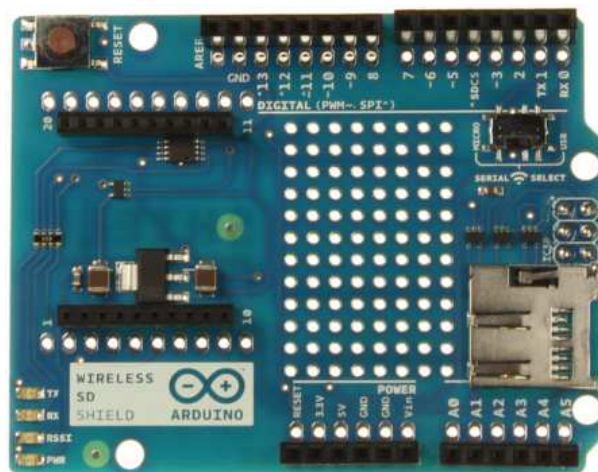


Figura 4: Wireless Shield

3.7 TARJETA DE RELAY

Manipulará el encendido y apagado del sistema, cuando el Arduino ADK lo indique. El Relay será capaz de soportar un paso de 250 voltios de corriente alterna y 10A, suficiente para controlar una válvula de tipo solenoide o bien una bomba de riego eléctrica.

3.8 BUZZER

Emitirá un sonido de alarma si el sistema enciende y no existe flujo de agua por el mismo.

3.9 LCD 16x2

Indicará los parámetros del sistema como: porcentaje de humedad del suelo, esta-tus del sistema (ON/OFF), alarma (ON/OFF), modo del sistema (automático/manual).

3.10 ANDROID

Es un sistema operativo desarrollado para dispositivos móviles de código abierto basado en Linux, que permite a cualquiera que quiera modificar o crear su propia versión del sistema. El lenguaje de programación base es C/C++ y Java. Hoy en día es el más utilizado en el mundo móvil y su éxito se debe en gran parte a la comodidad, flexibilidad y facilidad que les brinda a los desarrolladores de aplicaciones.

Anteriormente cuando se deseaba programar en el entorno Android era más complicado que en la actualidad, ya que necesitábamos instalar el ambiente por partes, hasta lograr la integración de las mismas. Hoy día es tan sencillo como dirigirnos a la sección de desarrolladores de Google y descargar el paquete llamado Android Development Tools (ADT). El mismo cuenta con todas las herramientas necesarias para comenzar en el mundo Android.

Todo lo anterior marca de cierta forma una estabilidad en este sistema operativo, en cuanto a una visión futura, ya que tanto la empresa creadora, como los desarrolladores alrededor del mundo aportan a que este se fortalezca y amplíe sus posibilidades.

En nuestra investigación utilizaremos Android para desarrollar una aplicación que denominaremos tipo servidor y otra denominada tipo usuario. Cada una tendrá su función específica, que más adelante explicaremos a detalle.

Vale destacar que utilizaremos dos servicios gratuitos que nos brinda Google, que son: Google Drive (repositorio virtual) y Google Spreadsheet (hoja de cálculo).

4. FUNCIONAMIENTO ESPERADO

Buscando el buen desempeño de la investigación, dividimos el desarrollo del prototipo de la siguiente manera: estructura del nodo principal, configuración de los nodos secundarios, integración de nodos y automatización, desarrollo de la aplicación servidor y usuario e integración de las partes.

4.1 ESTRUCTURA DEL NODO PRINCIPAL

Debemos lograr con nuestro Arduino Mega ADK las siguientes metas: automatización del sistema, comunicación con los nodos secundarios y el enlace con el dispositivo Android.

Básicamente el funcionamiento del nodo principal es solicitarle a cada nodo el valor de humedad del suelo que supervisa, luego dependiendo de la humedad promedio obtenida por todos los valores recopilados de los nodos, la rutina programada tomará la decisión si enciende o apaga el sistema. Si la decisión es encender el sistema, la rutina ordenará esta condición y se procederá a verificar el sensor de flujo en la línea de agua. Si no, se produce la circulación del líquido, entonces el sistema será apagado y el buzzer emitirá un sonido de alarma. La alarma también se mostrará en la pantalla LCD, para que de esta forma el operario pueda observar lo que está sucediendo. Con esta medida buscamos proteger el sistema de bombeo que es una de las partes más costosas en todo sistema de riego agrícola.

La transmisión y recepción del dato de humedad del suelo, se llevara a cabo gracias a los módulos XBee conectados en los nodos. Los mismos se configurarán en el modo más avanzado que nos brindan, llamado modo API. Debido a esto tendremos disponible la característica de direccionamiento y confirmación de paquetes. Por lo tanto nuestro nodo principal enviará una petición solicitando el valor de humedad a los nodos secundarios uno a la vez, quedando este a la espera primero de la recepción de la petición por parte del nodo secundario y luego a la espera del valor de humedad del suelo. Todo este proceso de petición, confirmación y recepción es repetido en cada nodo.

Se busca que la rutina programada en nuestro nodo principal esté dividida en dos partes: dispositivo móvil conectado y no conectado. Cuando exista un dispositivo conectado todas las variables del sistema serán enviadas al mismo. De esta forma el personal encargado podrá observar los datos del sistema, tanto en el dispositivo Android como en la pantalla LCD comentada anteriormente. Si sucede lo contrario, los datos solo serán mostrados en la pantalla LCD. Dividiendo la rutina en dos partes, pensamos lograr que el sistema sea autónomo y no dependa del dispositivo móvil para llevar a cabo la automatización.

A continuación se muestra en la Figura 5 el diagrama de flujo que se desea lograr en el nodo principal.

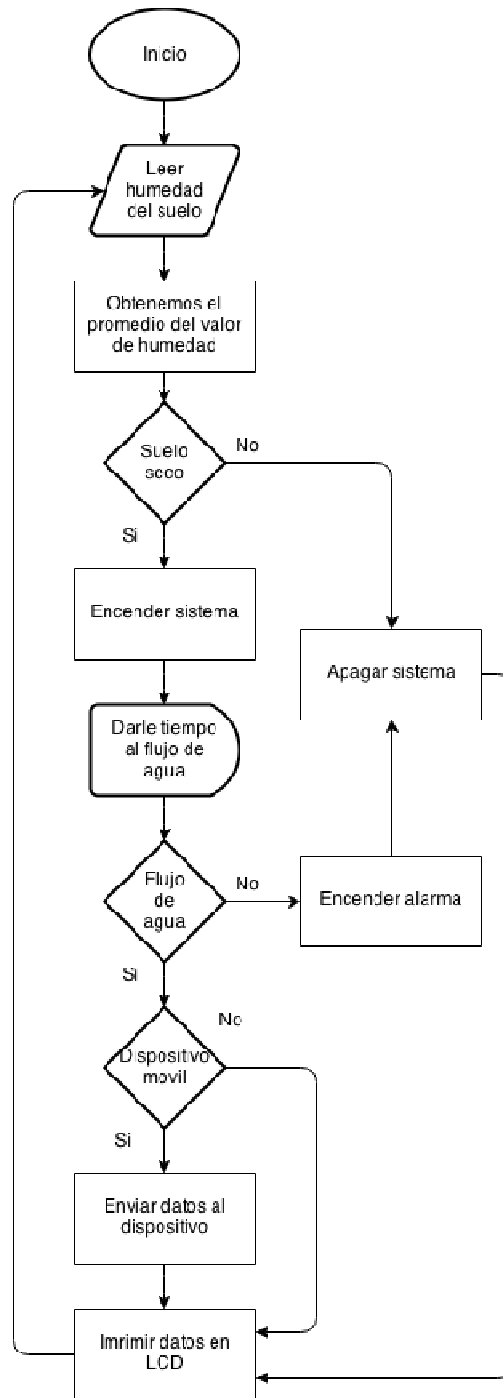


Figura 5: Diagrama de flujo esperado por la rutina del nodo principal

4.2 CONFIGURAR LOS NODOS SECUNDARIOS

Consistirán de un módulo XBee, del sensor Soil Moisture y Arduino Uno. Contemplándose la alimentación por celdas solares.

Los modulos XBee serán configurados de forma pasiva, de esta forma se ahorrará energía. Por lo tanto cada nodo secundario quedará a la espera de la petición del nodo principal. Cuando se obtenga una solicitud del nodo principal, el nodo secundario leerá a través del Soil Moisture Sensor el valor de la humedad del suelo, este a su vez será encapsulado y enviado al nodo principal, destacando que el nodo en discusión pasará por el proceso de recepción, confirmación y envío de paquete establecido por el modo API del módulo XBee.

En la Figura 6 se puede visualizar el funcionamiento buscado en la rutina de cada nodo secundario.

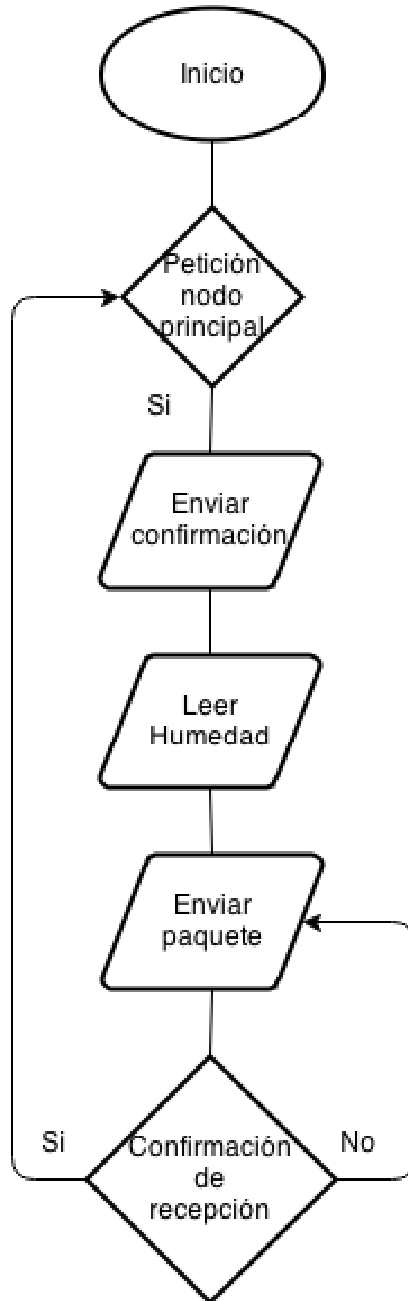


Figura 6: Rutina de los nodos secundarios

4.3 INTEGRACIÓN DE NODOS Y AUTOMATIZACIÓN

En esta etapa del desarrollo, buscamos unir, verificar y realizar pruebas con los nodos, para de esta forma confirmar la automatización del sistema. El esquema mostrado en la Figura 7, nos indica lo que queremos alcanzar.

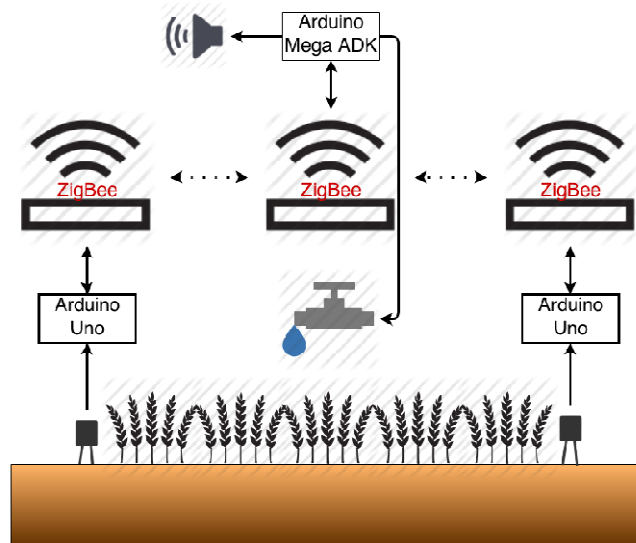


Figura 7: Esquema funcional integrando los nodos

4.4 DESARROLLO DE LA APLICACIÓN SERVIDOR

Deberá ser amigable al operario, mostrando los datos enviados por el Arduino Mega ADK, los cuales son: humedad promedio del suelo, estado del sistema (ON/OFF), alarma (ON/OFF) y modo del sistema (automático/manual). Además deberá subir estos datos en un proceso oculto al operario a un archivo Spreadsheet que nos brinda el servicio gratuito en la nube de Google Drive, con lo cual cumplimos el objetivo de poder monitorear el sistema de riego desde cualquier parte del mundo donde exista Internet y tengamos a la mano un dispositivo móvil. Bastará con ingresar a nuestra cuenta Google Drive y abrir el archivo correspondiente Spreadsheet.

Tendremos la opción de controlar el sistema de forma manual y poder encender, y apagar el riego.

Otra funcionalidad importante es servir de enlace con cualquier otro Android que tenga instalada la aplicación usuario, ya que a través de SMS el usuario enviara una petición de encender o apagar el sistema y la aplicación servidor debe ser capaz de transmitir esta solicitud al Arduino Mega ADK, por medio de un proceso o hilo que siempre estará a la espera de ello. Gracias a lo mencionado anteriormente, lograremos el control remoto del sistema.

4.5 APLICACIÓN USUARIO

Su funcionamiento básicamente consistirá en controlar el sistema de riego remotamente a través de SMS y visualizar el documento Spreadsheet en nuestra cuenta Google Drive. Para ello deberá el usuario otorgar los permisos para acceder a la cuenta Google Drive e ingresar el número celular del dispositivo servidor.

4.6 INTEGRACIÓN DE LAS PARTES

Una vez las partes relacionadas con Arduino sean estables y lo mismo suceda con las de Android, procederemos a unirlas entre ellas, siendo el punto de convergencia nuestro Arduino Mega ADK y la aplicación Android tipo servidor, enlazadas con las librerías proporcionadas por ambas plataformas, que se encuentran en el paquete llamado ADK 2012 Guide, que es el más reciente a la fecha.

Para el final del prototipo deseamos una estructura funcional como el que se muestra en la Figura 8.

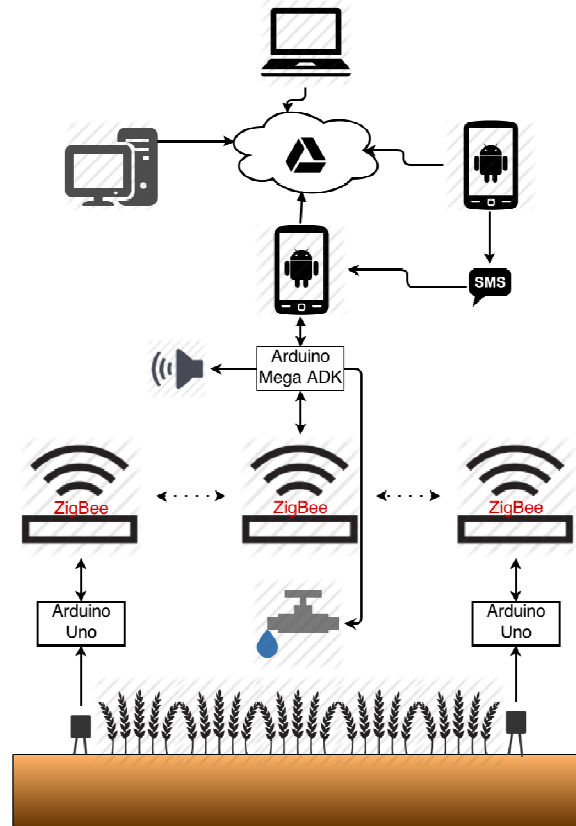


Figura 8: Esquema funcional del sistema

5. CONCLUSIONES

Deseamos obtener un sistema automatizado, estable y funcional que le brinde al agricultor comodidad, economía, seguridad y una mayor productividad.

Gracias a las herramientas de hardware, software y código abierto, tendremos una ventana abierta para futuras modificaciones e integraciones en la estructura base, aumentando los servicios y facilidades al usuario.

REFERENCES

- Böhmer, Mario. (2012). "Beginning Android ADK with Arduino". New York, Estados Unidos.
- Glaría, Jaime., Kouro, Samir. (2001). "Sensores de humedad". Universidad técnica Federico Santa María.
- Montesinos, José. (2013). "Red de sensores auto configurable mediante tecnologías ZigBee y Arduino con monitorización por aplicación Android". Tesis de grado. Universidad Politécnica de Cartagena, Cartagena, Colombia.
- Torres, Oscar. (2013). Arduino curso práctico de formación.
- Valverde, Rebaza, Jorge, Carlos. (2007). "El Estándar Inalámbrico ZigBee". Universidad Nacional de Trujillo, Trujillo, Perú.

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.