



Colegio Indoamericano S. C.

Sistema de riego automático y autosustentable

Clave de proyecto: CIN2016A50066

Autores:

María José Abarca Kerlegand

Fernando Jesús Rojas Reyes

Asesor:

Karen Ailed Neri Espinoza

Área: Áreas de Convergencia

Disciplina: Medio ambiente

Tipo investigación: Desarrollo tecnológico

Lugar: Tlalnepantla

Fecha: 19 de febrero de 2016

Resumen

El objetivo de esta investigación fue diseñar y construir un sistema de riego automático y autosustentable para azotea verde. Para comodidad y seguridad del responsable del cuidado de las plantas se implementó un sistema de envío de datos y notificaciones vía *bluetooth*: nivel de agua del depósito, nivel de humedad y momento del comienzo y fin del riego. Se utilizó una plataforma Arduino Mega para la programación de los sensores, válvula y *bluetooth*. También cuenta con un reloj de tiempo real para permitir el riego únicamente durante la noche, optimizando el riego.

El prototipo funciona correctamente ya que el sistema se activa cuando el nivel de humedad de la tierra se encuentra por debajo del rango preestablecido en la programación del Arduino. Si es de día se retrasa la abertura de la válvula del depósito de agua hasta que sea la hora establecida. El Arduino enviará la señal de cierre de la válvula cuando el nivel de humedad de la tierra sea el óptimo. Asimismo, el Arduino notificará el usuario vía *bluetooth* el nivel de humedad de la tierra, el momento de apertura y cierre de la válvula así como aquel en el cual el nivel de agua del depósito baje y sea necesario su llenado. Como mejora se pretende instalar un sensor de pH de la tierra para que notifique al responsable la necesidad de agregar un fertilizante.

Abstract

The objective of this research was to design and build a prototype of an automatic self-sustaining irrigation system for green roofs. For comfort and security, we implemented a system that sends data and notifications via Bluetooth. It informs of the water level in a tank, the moisture level and the hour of beginning/end of irrigation. Platform Arduino Mega was used for programming the sensors, valve and Bluetooth. The system also includes a clock connected to the platform and a program that enables irrigation only at night, optimizing water use.

The prototype works properly. The system activates when the soil moisture is below the pre-established value in the Arduino Mega program. If it is day time, the clock delays the opening of the valve until the established hour. Platform Arduino Mega sends the signal to close the valve when the soil moisture level is optimal. It also notifies the user, via Bluetooth, of the system status.

An intended improvement is to install a soil pH sensor to advise the user to add fertilizer.

Planteamiento del problema

¿Cómo poder mejorar el riego y manutención de azoteas verdes y jardines del hogar de una manera más económica, con mayor impacto en el cuidado del medio ambiente?

Cada vez los edificios van ganando terreno, empobreciendo el paisaje. Según el Consejo Mexicano de Edificación Sustentable, los edificios representan el 65% del consumo total de energía, emiten el 30% de gases de efecto invernadero y consumen el 35% del agua.

Es por eso, que es necesario buscar alternativas viables para reverdecer las áreas que se han perdido por la urbanización, y las azoteas y muros verdes son una alternativa real para contrarrestar este efecto, además que se ocupan espacios que están siendo desaprovechados.

Hipótesis

Si se crea un sistema de riego automático y autosustentable, se podrá hacer más eficiente y fácil el cuidado de las azoteas verdes.

Justificación

El sistema tiene como fin ayudar el mantenimiento de las plantas para atacar la principal problemática de las azoteas verdes y del medio ambiente.

Beneficios de las azoteas verdes:

- El espacio verde recuperado ayuda a purificar el aire y reducir los gases contaminantes en el entorno (disminuye el efecto invernadero)
- Reduce el nivel de ruido (son aislantes acústicos).
- Regula el clima local
- Aprovecha el agua de lluvia y la luz solar.
- Reduce las aguas de lluvias, anegaciones y contaminación del agua.
- Se convierte en un ecosistema para aves e insectos polinizadores.

Asimismo, el riego nocturno es más eficiente que el diurno ya que el agua que se evapora durante el riego por aspersión, modifica el microclima en el que se desarrollan las plantas; disminuye su temperatura, reduce su transpiración y aumenta el potencial hídrico de las hojas, lo que afecta su fisiología y producción porque disminuye su fotosíntesis.

Los sistemas de riego que existen en jardinería son:

- Riego con aspersores
- Riego con difusores
- Riego por goteo
- Riego subterráneo
- Riego con cintas de exudación: las cintas de exudación son tuberías de material poroso que distribuyen el agua de forma continua a través de los poros, lo que da lugar a la formación de una franja continua de humedad, que las hace muy indicadas para el riego de cultivos en línea.
- Riego con microaspersores: para textura arenosa son preferibles los microaspersores porque cubren más superficie que los propios goteros tradicionales, por ejemplo, para regar frutales.
- Riego con manguera
- Riego con regadera
- Macetas de autorriego
- Riego por surcos (por ejemplo, el huerto)
- Riego a manta (por ejemplo, inundando un arriate)

Objetivos

Objetivo general: El propósito de esta investigación fue diseñar y crear un sistema de riego automático para plantas en azotea verde.

Objetivos específicos:

- Programar una plataforma Arduino Mega para la abertura de la válvula sólo cuando la planta requiere agua y, además, ya no esté recibiendo la luz del sol.
- Instalar un indicador de nivel de agua en el depósito.
- Implementar al sistema riego comunicación vía *bluetooth* con el usuario.

Fundamentación teórica

Los componentes principales son: Arduino, que es una plataforma de licencia libre de fácil programación para prototipos, ya sean básicos como complejos, además de que con este microcontrolador se logra manejar hardware y software. El Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador. Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

Cuenta con una válvula solenoide. Los solenoides son muy útiles para realizar acciones a distancia sobre válvulas de control de gas y fluidos. Un solenoide es una bobina de material conductor cuyo funcionamiento se basa en campos electromagnéticos. Al pasar una corriente eléctrica a través de la bobina, se genera un campo electromagnético de cierta intensidad en el interior. Un émbolo fabricado de metal ferroso es atraído por la fuerza magnética hacia el centro de la bobina, lo que proporciona el movimiento necesario para accionar la válvula. La válvula se puede abrir o cerrar, no hay término medio, por lo que no se puede utilizar este sistema para regulación de flujos. Una vez que se activa el solenoide, la válvula se mantendrá abierta o cerrada, dependiendo del diseño, hasta que se corte la corriente eléctrica y desaparezca el campo electromagnético del solenoide. En este momento, un muelle o resorte empuja el émbolo de nuevo hacia su posición original cambiando el estado de la válvula. El hecho de que no se necesite manipulación física directa hace que las válvulas solenoides sean la mejor solución para controlar la entrada o salida de fluidos y gases en sitios de difícil acceso o dónde el entorno puede ser peligroso, como en sitios a altas temperaturas o con productos químicos peligrosos. Además, las bobinas del solenoide se puede cubrir con material ignífugo para hacerlas más seguras para ambientes peligrosos.

Una válvula de solenoide eléctrica sólo puede funcionar como dispositivo on/off y no puede ser utilizado para abrir o cerrar la válvula gradualmente en aplicaciones donde se requiera una regulación más precisa del flujo. En función del uso que se le va a dar a la válvula, se pueden utilizar bobinas capaces de trabajar de forma continua o en ciclos de duración determinada; siendo las de trabajo continuo normalmente más caras. Existen válvulas de solenoide aptas para su uso con corriente alterna, de 24 a 600 voltios, o para su uso con corriente continua, de 12 a 24 voltios.

Además se utilizó un sensor de humedad, el cual consiste en dos placas separadas entre sí por una distancia determinada. Ambas placas están recubiertas de una capa de material conductor. Si existe humedad en el suelo se creará un puente entre una punta y otra, lo que será detectado por un circuito de control con un amplificador operacional que será el encargado de transformar la conductividad registrada a un valor analógico que podrá ser leído por Arduino. De hecho, el circuito de control es el mismo que utiliza el módulo YL-83.

Metodología

Paso 1:

A una botella de plástico hacerle dos agujeros; uno en la parte inferior para poder introducir el agua y otro en la tapa para poder encajar la manguera.

Paso 2:

Poner la botella en un tubo de PVC de un metro y pegarla en el extremo del tubo para que quede fija y a una altura suficiente.



Paso 3:

Poner la electroválvula solenoide en el punto deseado del tubo y fijarla firmemente, medir la distancia de la válvula a la botella y cortar la manguera a dicha distancia.

Paso 4:

Adherir otro segmento de manguera al otro lado de la válvula y conectarla a las macetas, pegar el tubo a la base de acrílico.



Paso 5:

Colocar los paneles solares en la posición más favorable para su funcionamiento y conectarlas al Arduino Mega en donde irán conectados todos los componentes electrónicos.



Paso 6:

Conectar al Arduino los 4 sensores de humedad que estarán incrustados en la tierra de las plantas a regar.



Paso 7:

Unir a la válvula su fuente de energía independiente, la cual llevará una protección en el Arduino para que éste no se sobrecargue.

Paso 8:

Hacer la conexión del sensor de nivel de agua que irá en nuestro suministro de agua para que nos indique cuando el nivel esté por debajo del preestablecido y necesite rellenarse.

Paso 9:

Todas las conexiones de los componentes irán en la *Protoboard*, la cual junto con el Arduino será colocada fuera de la base de acrílico para su protección.

Paso 10:

Conectar junto al Arduino el *Real time clock module* que controla el tiempo en que se riegan las plantas y además, al tener una fuente de energía independiente, si el Arduino se queda sin energía éste no se desconfigura y no se pierden los datos programados.

Paso 11:

Pasar a la computadora, unir y programar cada aspecto de cada componente, y unificar el código. Conectar el *Bluetooth* junto con el Arduino para una mejor recepción de los mensajes que serán enviados al celular. El código que se diseñó fue el siguiente:

```
#include <Wire.h>

#include "RTCLib.h"

RTC_DS1307 RTC;

DateTime future = 0; // Variable para calcular cuando se debe desactivar la alarma

// CONFIGURACIÓN DE USUARIO

int PinLDR=A13;

int hour_alarm = 3; // Horas alarma. Formato 24h

int year_alarm = 2015; // Año alarma

int weekday_alarm1 = 1; // Dia de la semana alarma. L(1), M(2), MX(3), J(4), V(5), S(6), D(7)

int weekday_alarm3 = 3;

int weekday_alarm5 = 5;

int Duracion_RiegoAlbahaca=15000;
```

```

int Duracion_RiegoRocio=300000;

boolean all_hour = true; // Alarma para todas las horas

boolean all_weekday = true; // Alarma para todos los dias de la semana

boolean all_day = true; // Alarma para todos los dias

boolean all_month = true; // Alarma para todos los meses

boolean all_year = true; // Alarma para todos los años

int LedPin = 13;

int PinBomba=22;

int SensorHumeA2=A1;

int SensorHumeR2=A3;

int SensorHumeA1=A5;

int SensorHumeR1=A8;

int Sensornivel=A11;

void setup () {

Wire.begin(); // Inicia el puerto I2C

RTC.begin(); // Inicia la comunicaci3n con el RTC

RTC.adjust(DateTime(__DATE__, __TIME__)); // Establece la fecha y hora (Comentar una vez
establecida la hora)

////////////////////CONFIGURACION DE PINES////////////////////

pinMode(LedPin, OUTPUT); // Establece el pin LedPin como salida

pinMode(PinBomba,OUTPUT);

pinMode(SensorHumeA2,INPUT); //ALBAHACA 2

pinMode(SensorHumeR2,INPUT); //ROCIO 2

pinMode(SensorHumeA1,INPUT); //ALBAHACA 1

pinMode(SensorHumeR1,INPUT); //ROCIO 1

pinMode(A11,INPUT); //SENSOR DE AGUA

////////////////////////////////////

pinMode(Sensornivel,INPUT);

```

```

Serial.begin(9600); // Establece la velocidad de datos del puerto serie

} // LLAVE VOID SETUP

void loop(){

////////////////////////////////////TIEMPO////////////////////////////////////

DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC

Serial.print(now.year(), DEC); // Año

Serial.print('/');

Serial.print(now.month(), DEC); // Mes

Serial.print('/');

Serial.print(now.day(), DEC); // Dia

Serial.print(' ');

Serial.print(now.hour(), DEC); // Horas

Serial.print(':');

Serial.print(now.minute(), DEC); // Minutos

Serial.print(':');

Serial.print(now.second(), DEC); // Segundos

Serial.println();

////////////////////////////////////

////////////////////////////////////SENSORES HUMEDAD////////////////////////////////////

int HumedadA2=analogRead(SensorHumeA2);

int HumedadR2=analogRead(SensorHumeR2);

int HumedadA1=analogRead(SensorHumeA1);

int HumedadR1=analogRead(SensorHumeR1);

Serial.print("Humedad Albahaca 2:");

Serial.println(HumedadA2);

Serial.print("Humedad Rocio 2: ");

Serial.println(HumedadR2);

Serial.print("Humedad Albahaca 1: ");

```

```

Serial.println(HumedadA1);

Serial.print("Humedad Rocio 1: ");

Serial.println(HumedadR1);

////////////////////////////////////

if((HumedadA2>= 550)&&(HumedadR2>= 550)){
  if((HumedadA1>= 550)&&(HumedadR1>= 550)){
    if (year_alarm == now.year() || all_year ) {
      //if ((weekday_alarm1== now.dayOfWeek()) || all_weekday) {
      //if ((weekday_alarm3== now.dayOfWeek())|| all_weekday){
      //if ((weekday_alarm5== now.dayOfWeek())|| all_weekday){
      //if (hour_alarm == now.hour() || all_hour) {
        digitalWrite(PinBomba, HIGH);
        Serial.println("SE REGARA TU PLANTA DURANTE 15 s");
        delay(Duracion_RiegoAlbahaca);
      //} }}
    }
  }
  delay(1000); // La información se actualiza cada 1 seg.
}

analogRead(Sensornivel);

Serial.print("NIVEI DEL AGUA:");

Serial.println(analogRead(Sensornivel));

if (Sensornivel > 300)
{
  Serial.println("suministro lleno");
}
else {
  Serial.println("Sumistro vacio");
}

```

```
}  
int valorLDR= analogRead(PinLDR);  
while(valorLDR<250)  
{  
  digitalWrite(PinBomba,HIGH);  
  Serial.println("TU PLANTA SE ESTA REGANDO");  
  delay(500);  
  valorLDR= analogRead(PinLDR);  
}  
digitalWrite(PinBomba,LOW);  
delay(3000);  
}
```

Paso 12:

Ordenar todos los cables de las conexiones para una mejor presentación y añadir detalles de diseño o de escape de energía (si se presentan).



Paso 13:

Proteger al Arduino y al *Protoboard* para que en caso de lluvia o de algún fenómeno meteorológico éstos no se dañen.

Paso 14:

Probar que el código y todos los componentes funcionen correctamente en sincronía con el Arduino.

Paso 15:

Corregir errores (si los hay), en caso contrario el proyecto está listo.



Resultados

El dispositivo funcionó adecuadamente, ya que envía a la válvula la señal de apertura cuando el sensor de humedad registra tierra seca (siempre y cuando no haya luz solar, para optimizar el riego) y de cerrado cuando ha transcurrido el tiempo de riego determinado.

El sistema también envió los datos preestablecidos vía *bluetooth* al usuario.



Conclusiones

La instalación de este sistema ayuda a optimizar el tiempo y forma de riego en una azotea verde o jardín a un bajo costo y fácil manejo del mismo.

El sistema favorece a la disminución de temperaturas en hogares con azotea verde y la instalación de las mismas ayuda a disminuir los efectos de los gases invernaderos.

Referencias bibliohemerográficas y de Internet

- Bolton, W. (2010). Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica. México: Mc Graw Hill, 3ª Edición.
- Creus S., Antonio (1995). Instrumentación industrial. México: Alfaomega, Marcombo.
- Margolis Michael (1era Edición) (2011), *Arduino Cookbook*. California, EE. UU.: O'Reilly Media
- <http://www.instructables.com/id/Controlling-a-solenoid-valve-with-an-Arduino/>, recuperado el 19 de diciembre de 2015
- http://www.bibliotecapleyades.net/ciencia/ciencia_industryhealthiermedica144.htm, recuperado el 19 de diciembre de 2015
- <http://www.scielo.org.mx/pdf/agritm/v34n4/v34n4a9>, recuperado el 21 de diciembre de 2015
- <http://www.hunterindustries.com/es/homeowners/product-support/valves>, recuperado el 22 de diciembre de 2015