



Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura



Cátedra UNESCO
Tecnologías de apoyo para la Inclusión Educativa



REVISTA

JUVENTUD Y CIENCIA SOLIDARIA

En el camino de la investigación

DESARROLLO DE UN PROTOTIPO DE SISTEMA DE RIEGO AUTOMATIZADO PARA ZONAS URBANAS DE LA CIUDAD DE CUENCA

Eduardo Sebastián Sánchez Cordero y Marco Vinicio Jr. Sarmiento Añazco



Eduardo Sánchez Cordero, Soy estudiante de la Unidad Educativa Técnico Salesiano, sigo la especialidad de Mecatrónica, estoy cursando el Tercer Año de Bachillerato, nací el 14 de febrero del 2003, pertenezco a la selección de fútbol del colegio, pero mi preferencia siempre serán los estudios. Mi mayor pasión es estar en familia y amigos.



Marco Sarmiento Añazco, Soy estudiante de la Unidad Educativa Técnico Salesiano, sigo la especialidad de Mecatrónica y en este momento me encuentro cursando el tercer año de Bachillerato, nací el 20 de marzo del 2002, formo parte del Tecni Club y de la selección de fútbol de mi colegio. Tengo un gran apego a lo investigativo y lo que más me gusta es pasar tiempo con las personas que me rodean.

Resumen

Nuestro proyecto está conformado por varias estructuras tanto en el hardware como una electroválvula, un soporte, motor paso a paso, etc. y su software efectuado con Visual Basic y Arduino, esto con el objetivo de crear un sistema de riego ya sea a escala pequeña para un jardín en el patio de una casa hasta zonas extensas que conforman los regadíos de maizales, áreas verdes de colegios, etc. El prototipo de sistema de riego automatizado se realizó por ser práctico para las personas, dando comodidad en el hogar puesto que no se tendría que hacer un mayor esfuerzo por salir y regar sus plantas ya que lo harían mediante una aplicación. La idea surge a partir de la observación de cultivos y plantas marchitadas por falta de agua,

esto normalmente se da por la falta de tiempo que se tiene en el día a día o por no recordar la presencia de las plantas. Como se especificó en el punto uno, nos hemos enfocado en zonas domésticas, es decir, a escalas pequeñas como jardines, áreas de trabajo, de oficina, etc. La funcionalidad se basa en poder regar plantas en un determinado tiempo, en nuestro caso hemos incluido la función de escoger cuanta cantidad de agua se desea regar. La cantidad de agua se relaciona con la cantidad de tiempo que se mantiene abierta la electroválvula.

Palabras clave: Riego, plantas, control, desarrollo productivo.

1. Explicación del tema

La funcionalidad se desarrolló mediante un Arduino, el ingreso de los datos era mediante la aplicación Visual Basic en donde podía poner un determinado tiempo para regar las plantas, en nuestro caso hemos incluido la función de escoger la cantidad de agua que se desea regar. Ésta última operación tiene que ser estudiada con la cantidad de mililitros que salen por segundo al mantener abierta la electroválvula.

“Las plantas para poder vivir necesitan de casi los mismos elementos que los seres humanos, si se les eliminara tan sólo uno de ellos, mueren. Estos elementos son: tierra, aire, agua y luz” (Matute, 2015, pág. 1). Este es uno de los factores que nos ha impulsado a crear dicho proyecto ya que también se debe tener en cuenta la población que podría beneficiarse de nuestro producto, así como familias, empresarios, agricultores, etc.

Los materiales a utilizar fueron escogidos cuidadosamente ya que al hacer un estudio adecuado de lo que necesitábamos debíamos saber el funcionamiento de cada aparato o elemento que se iba a comprar, aparte se hicieron bocetos que ayudaron en la construcción del proyecto. Los materiales utilizados en la maqueta se muestran en la Tabla 1.

Tabla 1. Materiales utilizados en la maqueta

Materiales	Precio
Electroválvula	\$8.00
Abrazaderas	\$0.25
Manguera (metro)	\$1.00
Adaptador de manguera	\$0.50
Sensor de humedad para tierra	\$9.50
LCD	\$7.50
Módulo relé	\$7.00
Motor paso a paso	\$5.00
Driver	\$2.50
Arduino	\$14.50

Comenzamos a ver el funcionamiento de cada uno de los aparatos y cómo funcionaban en conjunto, decidimos crear el primer prototipo de la funcionalidad de la electroválvula junto con el relé, debemos especificar que este módulo estaba siendo controlado mediante

el Arduino y Visual Basic, los primeros resultados fueron los esperados y conseguimos hacer funcionar la electroválvula por primera vez.

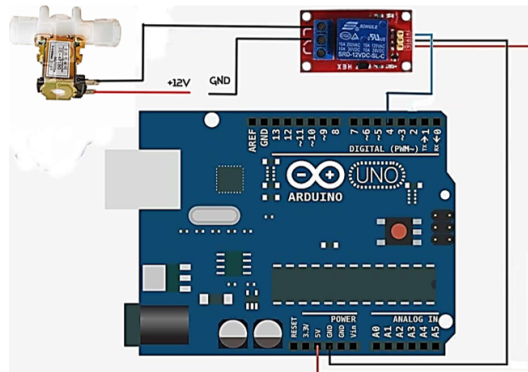


Figura 1. Conexión de la electroválvula, Arduino y relé. Elaboración propia

El movimiento del motor se consiguió con el Driver que es un elemento de la mecánica que “sirve principalmente para tener el control de los motores, este es compatible con cualquier elemento electrónico que permita un flujo mayor a 5 voltios” (Geek, 2018).

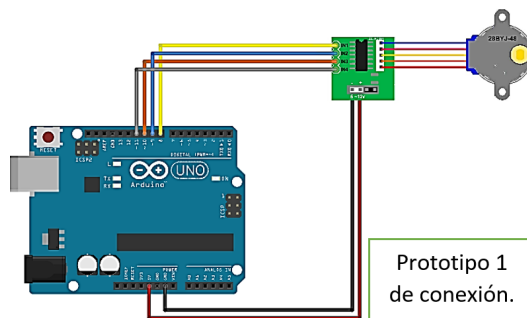


Figura 2. Conexión del motor paso a paso, Arduino y el Driver del motor. Elaboración propia

En esta parte utilizamos la librería LiquidCrystal (Para la Lcd) en donde también definimos la fila, y los colores, utilizamos los pines para la Lcd (7, 6, 5, 4, 3, 2), así como para el motor definimos los pines (8, 9, 10, 11) y creamos una variable de tipo byte y otras de tipo int.

```

#include <LiquidCrystal.h> // libreria
#define COLORES 16 // color 16 del lcd
#define FILA 2 // cuantas filas hay
byte a;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // pines
const int B = A0; // pin analogico
int C, T, Q, P; // constantes
int bobina1 = 8;
int bobina2 = 9;
int bobina3 = 10;
int bobina4 = 11;
int tiempo = 2; // No puede ser uno ya q el minimo de tiempo a definir es 2

void setup() {
  Serial.begin(9600);
  pinMode(2, INPUT);
  lcd.begin(COLORES, FILA);

  pinMode(bobina1, OUTPUT); // todos los pines como salida
  pinMode(bobina2, OUTPUT);
  pinMode(bobina3, OUTPUT);
  pinMode(bobina4, OUTPUT);
}

void loop()
{
  int A = analogRead(B);
  lcd.clear(); // Debe comenzar limpio
  if (A >= 1021) // Datos del Lcd donde 1021 seria lo maximo (tierra seca)
  {
    lcd.clear();
    lcd.setCursor(0, 0); // (x, y)
    lcd.print("Proyect Jr y Edu ");
    lcd.setCursor(0, 1);
    lcd.print("Tierra Seca!!");
  }
  else
  {
    C = A*100;
    P = { C/(323)};
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Proyect Jr y Edu ");
    lcd.setCursor(0, 1);
    lcd.print("Humedad: ");
    lcd.setCursor(9, 1);
    lcd.print(P);
    lcd.setCursor(11, 1);
    lcd.print("°");
  }

  delay(5000); // Actualizacion del lcd
}

```

Figura 3. Inicio del programa Motor conjunto a la Lcd. Elaboración propia

```

int A = analogRead(B);
lcd.clear(); // Debe comenzar limpio
if (A >= 1021) // Datos del Lcd donde 1021 seria lo maximo (tierra seca)
{
  lcd.clear();
  lcd.setCursor(0, 0); // (x, y)
  lcd.print("Proyect Jr y Edu ");
  lcd.setCursor(0, 1);
  lcd.print("Tierra Seca!!");
}
else
{
  C = A*100;
  P = { C/(323)};
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Proyect Jr y Edu ");
  lcd.setCursor(0, 1);
  lcd.print("Humedad: ");
  lcd.setCursor(9, 1);
  lcd.print(P);
  lcd.setCursor(11, 1);
  lcd.print("°");
}

delay(5000); // Actualizacion del lcd
}

```

Figura 4. Programación que se hizo de la Lcd. Elaboración propia

```

}
delay(5000); // Actualizacion del lcd

a = Serial.read();
{
  if (a==99)
  {
    for (int i = 0; i < 20000; i++) // 20000*4 = 80000 pasos
    {
      digitalWrite(bobina1, HIGH);
      digitalWrite(bobina2, LOW);
      digitalWrite(bobina3, LOW);
      digitalWrite(bobina4, LOW);
      delay(tiempo);

      digitalWrite(bobina1, LOW);
      digitalWrite(bobina2, HIGH);
      digitalWrite(bobina3, LOW);
      digitalWrite(bobina4, LOW);
      delay(tiempo);

      digitalWrite(bobina1, LOW);
      digitalWrite(bobina2, LOW);
      digitalWrite(bobina3, HIGH);
      digitalWrite(bobina4, LOW);
      delay(tiempo);

      digitalWrite(bobina1, LOW);
      digitalWrite(bobina2, LOW);
    }
  }
}

```

Figura 5. Programación que se hizo del motor. Elaboración propia

```

digitalWrite(bobina3, LOW);
digitalWrite(bobina4, HIGH);
delay(tiempo);
}

for (int i = 0; i < 20000; i++) // 20000*4 = 80000 pasos de regreso
{
  digitalWrite(bobina1, LOW);
  digitalWrite(bobina2, LOW);
  digitalWrite(bobina3, LOW);
  digitalWrite(bobina4, HIGH);
  delay(tiempo);

  digitalWrite(bobina1, LOW);
  digitalWrite(bobina2, LOW);
  digitalWrite(bobina3, HIGH);
  digitalWrite(bobina4, LOW);
  delay(tiempo);

  digitalWrite(bobina1, LOW);
  digitalWrite(bobina2, HIGH);
  digitalWrite(bobina3, LOW);
  digitalWrite(bobina4, LOW);
  delay(tiempo);

  digitalWrite(bobina1, HIGH);
  digitalWrite(bobina2, LOW);
}

```

Figura 6. Programación que se hizo del motor. Elaboración propia

```

digitalWrite(bobina2, LOW);
digitalWrite(bobina3, HIGH);
digitalWrite(bobina4, LOW);
delay(tiempo);

digitalWrite(bobina1, LOW);
digitalWrite(bobina2, HIGH);
digitalWrite(bobina3, LOW);
digitalWrite(bobina4, LOW);
delay(tiempo);

digitalWrite(bobina1, HIGH);
digitalWrite(bobina2, LOW);
digitalWrite(bobina3, LOW);
digitalWrite(bobina4, LOW);
delay(tiempo);
}
}
if (a==100)
{
  digitalWrite(bobina1, OUTPUT);
  digitalWrite(bobina2, OUTPUT);
  digitalWrite(bobina3, OUTPUT);
  digitalWrite(bobina4, OUTPUT);
}
}

```

Figura 7. Programación que se hizo del motor. Elaboración propia

Utilizamos el Relé para que controle la electroválvula ya que su función es la de cerrar o abrir un aparato electrónico, en este caso la electroválvula. Cabe recalcar que esta parte lo aislamos utilizando un Arduino solo para esta sección, entonces utilizamos el pin (2), definimos una variable tipo byte, a continuación, decidimos que siempre debe empezar apagado, también utilizamos código ASCII.

En Visual Basic se controla el paso de agua, teniendo dividido el programa en 2 partes:

- La manual: El usuario riega las plantas abriendo y cerrando la electroválvula mediante la aplicación.
- La automática: El usuario configura cada cuanto tiempo y que cantidad de agua debe regar la elec-

trovávula, dando la facilidad que ya configurado el sistema valdrá por cuenta propia.

```

int relay = 2; // el pin del Reley
byte a; // Como constante
void setup() {
  pinMode(relay,OUTPUT); //Comenzar Apagado
  Serial.begin(9600);
}
void loop() {
  a = Serial.read(); //Lectura
  {
  if(a==97) //Codigo ascii (a)
  {
    digitalWrite(relay, HIGH);
    Serial.println("Relay accionado");
  }
  if (a==98) //Codigo ascii (b)
  {
    digitalWrite(relay, LOW);
    Serial.println("Relay no accionado");
  }
  }
}

```

Figura 8. Programación que se hizo del Relé. Elaboración propia

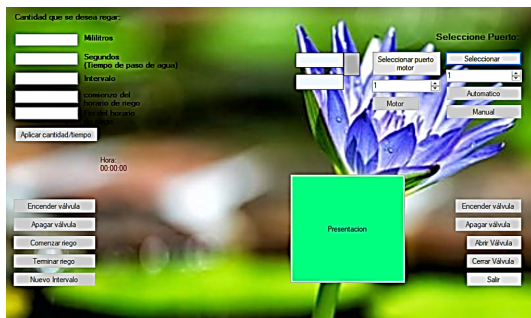


Figura 9. Interfaz de Visual Basic. Elaboración propia

Incorporación del mecanismo de riego con el sistema mediante el Arduino.

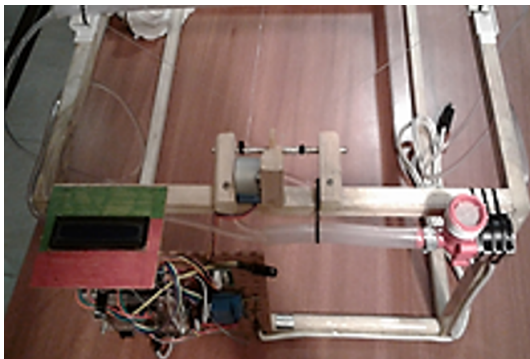


Figura 10. Se muestra todas las conexiones realizadas ya incorporadas en la maqueta. Elaboración propia

Se realizó pruebas de riego en zonas con macetas, jardines personales, comprobando la utilidad y lo factible que puede ser, demostrando lo fácil, sencillo y

seguro de utilizar dentro de los hogares mejorando la distribución del agua de una manera que nos permite ahorrar la misma.

Conclusiones

El prototipo del sistema de riego automatizado fue una de las mejores ideas ya que aparte de ser un trabajo bien elaborado y teniendo la capacidad de generar una comodidad a las personas ya siendo una maqueta, también tuvo el apoyo de muchas personas que se interesaron ya sea por el mecanismo o sus elementos de fabricación.

Lanzar este proyecto al mercado sería muy viable, ya que sería aceptado en las familias, empresas, etc. Las personas que tenían terrenos extensos en donde querían apostar por una nueva idea, veían a nuestro proyecto muy prometedor.

En este proyecto se demostró la capacidad de utilizar recursos como Arduino, Visual Basic para generar a través de una maqueta una ayuda a la hora de regar las plantas y además de ver la humedad en la tierra. Con la interacción que nos propone Visual Basic al momento de que se ingresan los datos nos permitió generar esa comodidad de que al hacer un clic se pueda realizar dicha acción, generando así una satisfacción a nosotros como a las personas que lo utilizaron.

Bibliografía

- Geek, B. (15 de Diciembre de 2018). Controlador Easy-Driver para motores paso a paso. Recuperado el 4 de Diciembre de 2018, de BricoGeek: <https://bit.ly/364Cc4v>
- Geek, B. (15 de Diciembre de 2018). Sensor de humedad del suelo. Recuperado el 15 de Diciembre de 2018, de BricoGeek: <https://bit.ly/34XL5MY>
- Mecafenix, F. (20 de Abril de 2017). Motor paso a paso ¿que es y como funciona? Recuperado el 5 de Diciembre de 2018, de Ingeniería mecafenix: <https://bit.ly/2rkgVF5>
- Yubal. (3 de Agosto de 2018). Qué es Arduino, cómo funciona y qué puedes hacer con uno. Recuperado el 18 de Noviembre de 2018, de Xataka Basics: <https://bit.ly/2LpsUrK>