

Proyecto: Sistema de Riego Automático

Automatización con Arduino para el depósito de agua para riego inteligente

Objetivo 1: Maquetar y Diseñar un Sistema de Riego Automático

Objetivo 2: Programar la Placa Arduino para automatización del Riego de una Huerta Escolar.

Destinatarios: Alumnos de toda la Escuela, Escuela en general.

Responsables:

- Directora Prof. Marisa Arrieta
- Micaela Jaiff - Asesora Pedagógica
- Gustavo Nuñez – Ref. Técnico

Idea del Proyecto de Integración

Se integran en las tareas, El Proyecto de huerta Escolar y el Proyecto de Taller de Robótica Educativa

<https://www.atilioanastasi.edu.ar/proyectos>



Proyecto Robótica Educativa

Aquí encontrarás info, fotos y recursos de nuestro Proyecto de Robótica Educativa 2023-2024 🤖



Proyecto Huerta Escolar

Aquí encontrarás info, fotos y recursos de nuestro Proyecto de Huerta Escolar 🌱

Los Alumnos integrarán conceptos, ideas y prácticas del **Proyecto Huerta Escolar** para la realización de una maqueta y prototipo funcional de Riego Automatizado utilizando componentes eléctricos y electrónicos.

En el desarrollo del Proyecto, darán sus primeros pasos en “Pensamiento Computacional, “Trabajo Colaborativo y “Pensamiento Crítico” para obtener un producto funcional.

Evaluación

Se realizarán actividades prácticas evaluables con el método de Observación directa en el Taller de Robótica Educativa, a realizarse con una frecuencia semanal de 2 clases.

Referencia_Proyecto

<https://drive.google.com/file/d/1539OPZxQMvFA3GE87lrOFvPZujzvuHb1/view>

<https://github.com/gustavon2023/Arduino/tree/main/Ejercicios%20Taller%20RE%20Ok/Proyecto-Sistema-Riego-Simple>

<https://github.com/gustavon2023/Arduino/tree/main/Ejercicios%20Taller%20RE%20Ok/proyecto-bomba-agua2>

<https://github.com/gustavon2023/Arduino/tree/main/Ejercicios%20Taller%20RE%20Ok/proyecto-sensor-humedad>

<https://github.com/gustavon2023/Arduino/tree/main/Ejercicios%20Taller%20RE%20Ok/Sistema-Riego-DHT11>

Tabla de contenidos

- Materiales para realizar el Proyecto
- Video explicación proyecto.
- Esquema de conexiones Arduino para su construcción
- Código de Arduino para la programación.
- Pruebas y Calibración

Materiales Necesarios para su Construcción

A continuación veremos los diferentes materiales que se van a utilizar para realizar nuestra grúa con Arduino y una breve descripción de los mismos.

Protoboard: Tabla con orificios (pines) la cual está conectada internamente y usaremos para realizar nuestras conexiones para el proyecto.

Maceta o Huerta

Placa de Arduino UNO: Es el cerebro de nuestro proyecto, encargada de controlar todos los procesos del mismo mediante el código que encontrarás más adelante.

Mini-Bomba sumergible: Bomba de agua 12 volt

Sensor Humedad resistivo de suelo o tierra

Sensor DHT-22 (opcional)

Pantalla LCD I2C (Opcional)

Resistencia 4.7 K; 0.2 K

Un módulo relay o relé de un canal

Tacho o depósito plástico para Agua

Pulsador

Fuente de 5-9 volt

Manguera plástica transparente (2 mts)

Video explicación proyecto grúa con Arduino

A continuación, se muestra el video con dicho contenido, mucho mas visual y fácil de comprender

Link: <https://www.youtube.com/watch?v=odD9rxIVZns>

Opcional: <https://www.youtube.com/watch?v=IF9qlw6TVJI>



Actividades

Calibrar sensores de Humedad de Piso

1.- Una vez escrito el código Arduino para la utilización de un Sensor de Humedad YL-69

a.- Medición # 1 con Suelo Seco

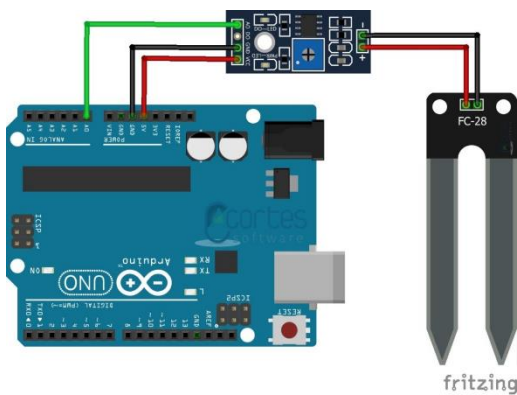
a1.- Tomar valores con el Monitor Serie

Para realizar esta medición, utilizar tierra seca en un vaso plástico o masetta

b.- Medición # 2 con Suelo Húmedo

b1.- Tomar valores con el Monitor Serie

Para realizar esta medición, utilizar tierra húmeda en un vaso plástico o masetta



```
prueba-sensor-humedad Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

prueba-sensor-humedad

// Prueba de funcionamiento del sensor de humedad HW-080 con Arduino UNO

// Pines
const int pinAnalog = A0; // AO del sensor al pin A0 del Arduino
const int pinDigital = 7; // DO del sensor al pin digital 7 del Arduino
const int pinLed = 13; // LED integrado en la placa Arduino UNO

void setup() {
  Serial.begin(9600);
  pinMode(pinDigital, INPUT);
  pinMode(pinLed, OUTPUT);
  Serial.println("Prueba de Sensor de Humedad HW-080 iniciada...");
}

void loop() {
  // Leer valores
  int valorAnalogico = analogRead(pinAnalog);
  int valorDigital = digitalRead(pinDigital);

  // Mostrar valores en el monitor serie
  Serial.print("Valor Analógico: ");
  Serial.print(valorAnalogico);
}
```

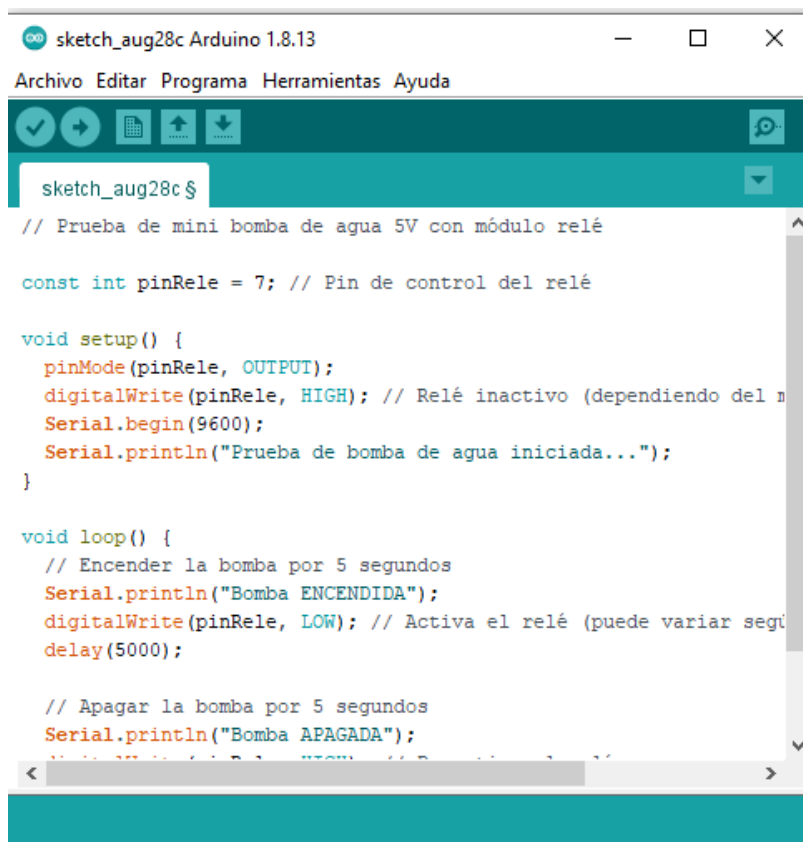
Guardado.

Prueba y funcionamiento de la Mini Bomba de Agua sumergible 5 volt

Puesta en Funcionamiento

Utilizaremos para la puesta en funcionamiento un pulsador, un Relé de 5 volt y una fuente de 5 volt. También un recipiente plástico para almacenamiento o repositorio de Agua.

Para conectar una mini bomba de agua a Arduino, necesitas un módulo [relé](#) porque la bomba consume más corriente de la que Arduino puede suministrar directamente. Conecta la fuente de alimentación de la bomba a los pines del relé y el relé a un pin digital de Arduino. Luego, conecta el terminal positivo de la bomba al terminal "NO" (normalmente abierto) del relé, el terminal negativo de la bomba al negativo de la fuente, y la tierra de la fuente de alimentación al GND de Arduino



```

sketch_aug28c Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

sketch_aug28c $
// Prueba de mini bomba de agua 5V con módulo relé

const int pinRele = 7; // Pin de control del relé

void setup() {
  pinMode(pinRele, OUTPUT);
  digitalWrite(pinRele, HIGH); // Relé inactivo (dependiendo del m
  Serial.begin(9600);
  Serial.println("Prueba de bomba de agua iniciada...");
}

void loop() {
  // Encender la bomba por 5 segundos
  Serial.println("Bomba ENCENDIDA");
  digitalWrite(pinRele, LOW); // Activa el relé (puede variar segú
  delay(5000);

  // Apagar la bomba por 5 segundos
  Serial.println("Bomba APAGADA");
  digitalWrite(pinRele, HIGH);
  delay(5000);
}
  
```



X2

Bomba sumergible que funciona con 5 v.

Relé



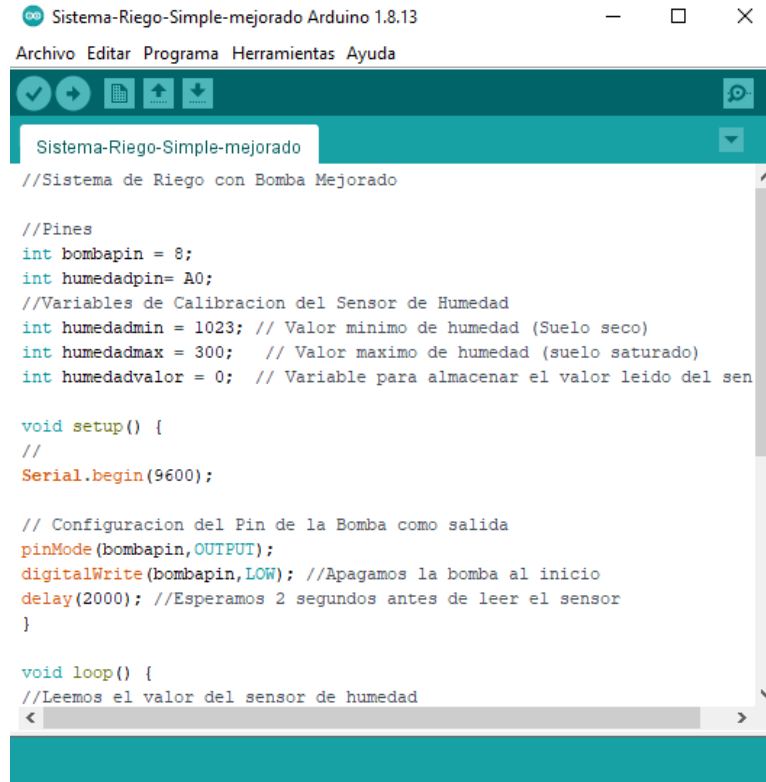
Esquema de conexiones Arduino

Una de las partes más importantes a la hora de realizar nuestro montaje pasa por conectar correctamente los diferentes elementos. Es fundamental que los componentes estén correctamente conectados.

Con el fin de evitar errores de montaje o conexiones erróneas, a continuación, te dejamos el esquema de conexiones empleado para este proyecto. Con este esquema de conexiones es posible usar el código que puedes encontrar al final del post sin necesidad de hacer ninguna modificación.



Código de Arduino Final



The screenshot shows the Arduino IDE interface with the project name 'Sistema-Riego-Simple-mejorado' and version '1.8.13'. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. The toolbar contains icons for opening, saving, and running the code. The code editor displays the following code:

```
//Sistema de Riego con Bomba Mejorado

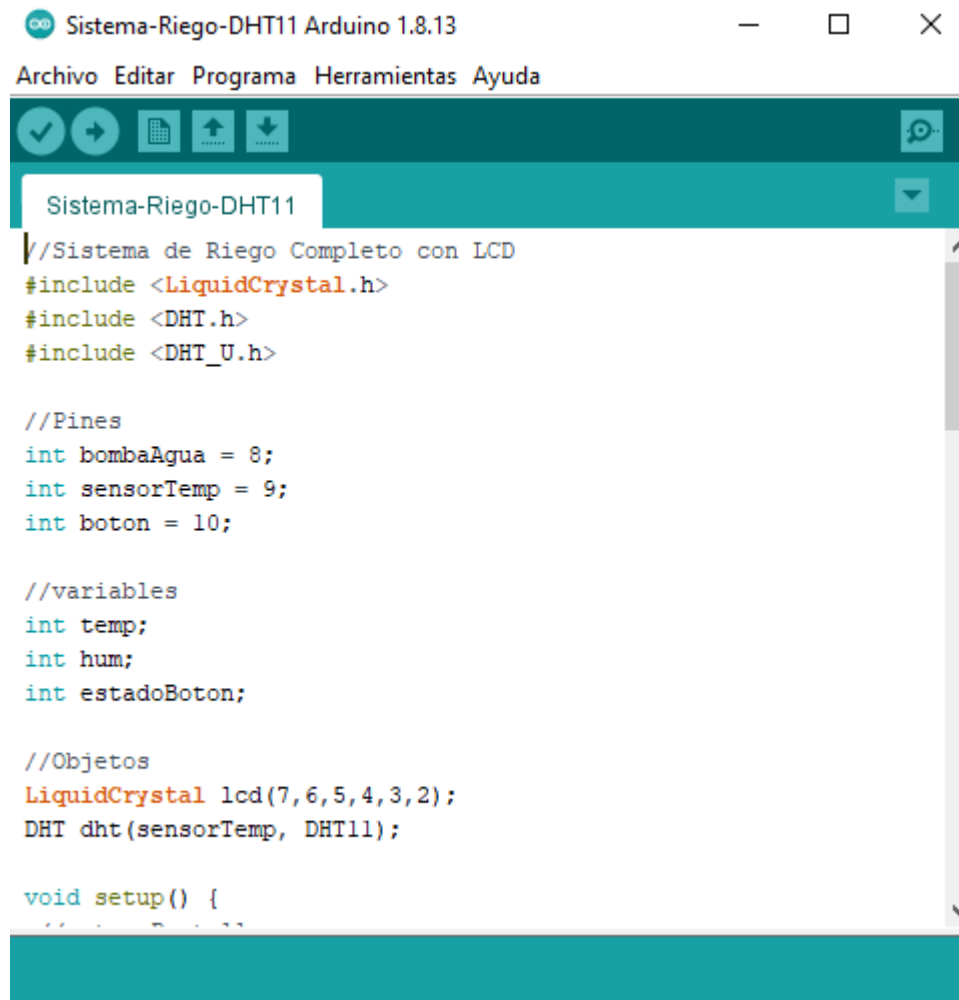
//Pines
int bombapin = 8;
int humedadpin= A0;
//Variables de Calibracion del Sensor de Humedad
int humedadmin = 1023; // Valor minimo de humedad (Suelo seco)
int humedadmax = 300; // Valor maximo de humedad (suelo saturado)
int humedadvalor = 0; // Variable para almacenar el valor leído del sen

void setup() {
//
Serial.begin(9600);

// Configuración del Pin de la Bomba como salida
pinMode(bombapin,OUTPUT);
digitalWrite(bombapin,LOW); //Apagamos la bomba al inicio
delay(2000); //Esperamos 2 segundos antes de leer el sensor
}

void loop() {
//Leemos el valor del sensor de humedad
```


Código de Arduino Opcional



The screenshot shows the Arduino IDE interface. The title bar reads "Sistema-Riego-DHT11 Arduino 1.8.13". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for checking, running, uploading, and downloading. The file explorer on the left shows a single file named "Sistema-Riego-DHT11". The main text area contains the following code:

```
//Sistema de Riego Completo con LCD
#include <LiquidCrystal.h>
#include <DHT.h>
#include <DHT_U.h>

//Pines
int bombaAgua = 8;
int sensorTemp = 9;
int boton = 10;

//variables
int temp;
int hum;
int estadoBoton;

//Objetos
LiquidCrystal lcd(7,6,5,4,3,2);
DHT dht(sensorTemp, DHT11);

void setup() {
```