



## **INFORME CONTROL DE RIEGO**

Fecha: 14/05/22

José Luis Campero Romero

1. Introducción
2. Objetivos
3. Calibración
4. Caso 1 : Arduino uno
5. Caso 2: Arduino ESP8266
6. Caso 3: Arduino ESP8266 con caudalímetro
7. Firebase + AppInventor
8. Conclusiones
9. Bibliografía

### **1.- Introducción**

Con este trabajo pretendemos realizar una gestión inteligente de riego controlando la humedad del suelo con un sensor.

Los datos obtenidos del Arduino se transmiten a través de una tarjeta ESP8266 a una base de datos en la nube FIREBASE para su control desde una App para Smartphone.

Página de descargas:

<https://sites.google.com/view/controlriego/home>

## **2.- Objetivos**

Seguimiento del agua en el suelo para evitar el estrés hídrico, evaluando el drenaje, caudal y la programación del riego de forma automática y manual.

## **3. Calibración**

Utilizo un sensor de low-cost DFROBOT para medir la humedad y tras la calibración de dicho sensor obtengo los siguientes datos:

0-300 suelo seco.

300-700 humedad media

700-900 agua pura.

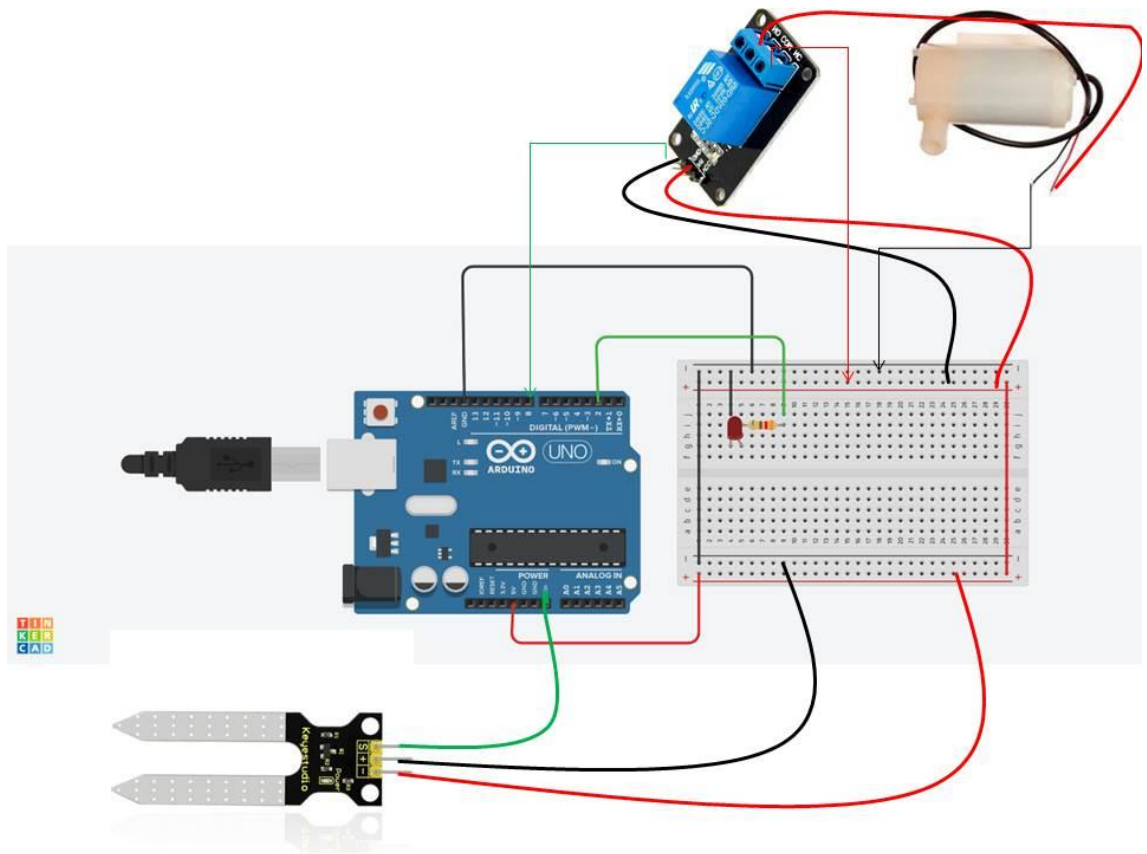
Según esta calibración regaremos cuando el sensor se encuentre entre los valores 0-300 que nos indica suelo seco.

## **4.- Caso 1 : Arduino UNO**

El Relé se conecta al pin 8, el Led al pin 7 digital del Arduino UNO y el sensor de humedad va conectado a la puerta A0 analógica del mismo Arduino.

El relé conmuta cuando el valor del sensor de humedad se encuentra entre los valores de 0 a 300 ya que con esos valores consideramos que la tierra está seca y necesita agua para no causar estrés hídrico en el cultivo. De esta forma se pone en funcionamiento la bomba y comienza a regar, el led se enciende para indicarnos que el riego se está realizando.

Las conexiones las podemos ver en la figura siguiente:



## 5. Caso 2: Arduino ESP8266

Utilizamos el ESP8266 para poder conectarnos a internet y pasar los datos del arduino a una base de datos en la nube, en nuestro caso utilizamos FIREBASE de Google. Los datos almacenados serán controlados desde una app desarrollada para Android con AppInventor desde donde visualizaremos los datos obtenidos desde el Arduino, de esta forma podremos ver si nuestro control de riego está regando o no. También en función de la información obtenida, el usuario final podrá tomar la decisión de regar o parar el riego según decida.

Para la conexión con FIREBASE he tenido que instalar 2 librerías en el software del Arduino:

Librerías de Firebase descargadas desde:

<https://github.com/FirebaseExtended/firebase-arduino>

Librería ArduinoJson version 5.13.5

Al conectar el relé con el ESP8266 no cambiaba de valor para activar el conmutador. Cuando ponía el pin del relé en HIGH, mantenía siempre el mismo valor a "0" debido a que el ESP8266 trabaja con 3,3v y el relé necesita 5v. Para conseguir esto he utilizado otro panel conectado con el Arduino UNO que le pasa los 5v. A pesar de esto seguía sin poder conmutar el relé para

accionar la bomba y he tenido que utilizar un transistor para que me haga la función de conmutador.

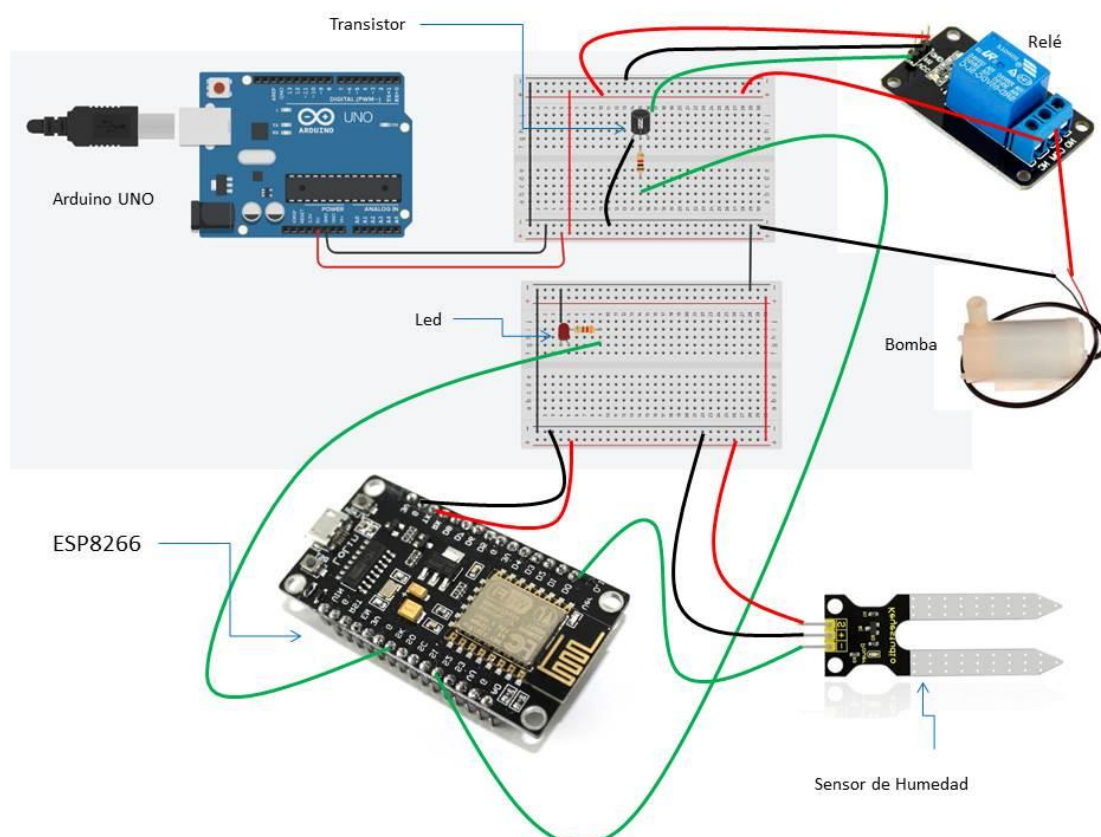
Tengo que conectar el GND de una placa con el GND de la otra. La pata izquierda del transistor a un GND, la pata derecha la conectamos con la señal del relé y a la pata central le conectamos una resistencia y luego ésta a la señal D4 de control de la tarjeta ES8266.

El Led se conecta a través de una resistencia al pin D7 digital del ESP8266 y el sensor de humedad va conectado a la puerta A0 analógica del mismo Arduino. Este sensor de humedad es alimentado a través del ESP8266 ya que con 3,3v tiene suficiente.

El relé conmuta cuando el valor del sensor de humedad se encuentra entre los valores de 0 a 300 ya que con esos valores consideramos que la tierra está seca y necesita agua para no causar estrés hídrico en el cultivo. De esta forma se pone en funcionamiento la bomba y comienza a regar, el led se enciende para indicarnos que el riego se está realizando.

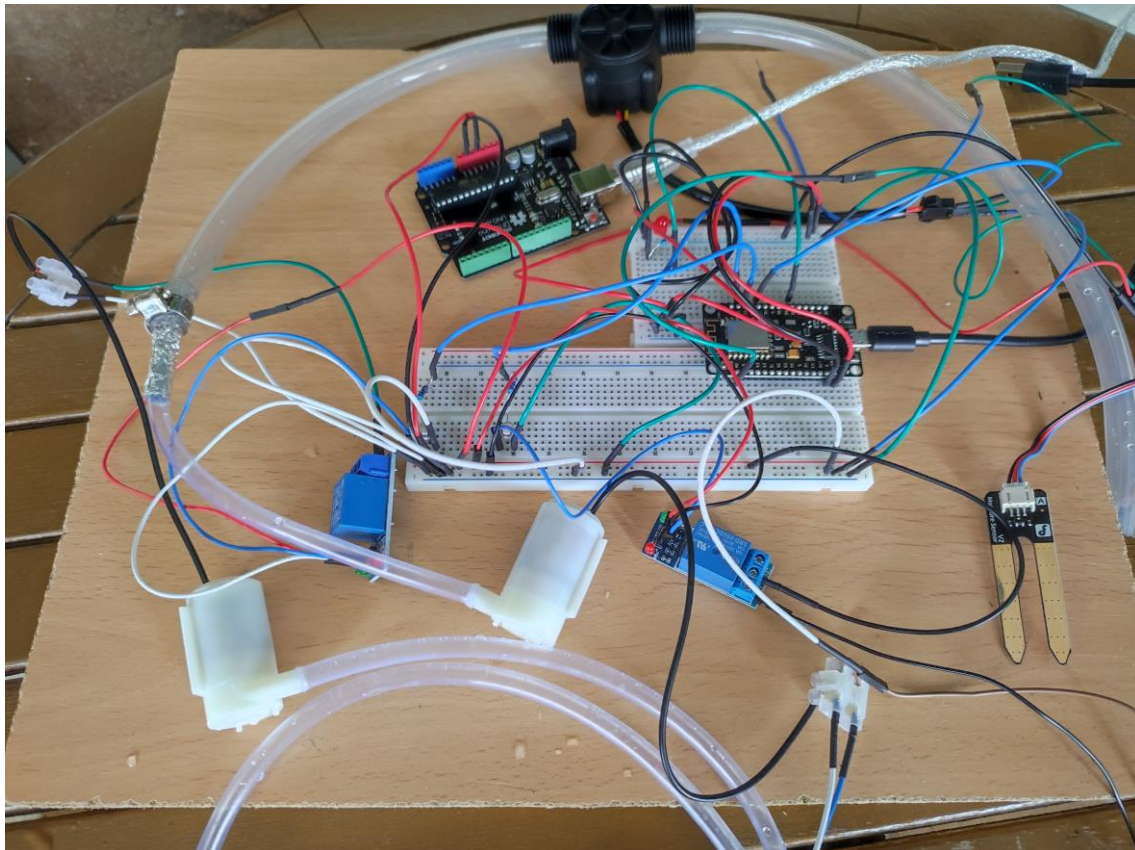
Desde la app podemos visualizar desde cualquier lugar los valores que estamos recogiendo del sensor y la situación del riego pudiendo manipular la situación manualmente. Al pulsar el botón de riego activamos o desactivamos el riego.

Las conexiones las podemos ver en la figura siguiente:



## 6. Caso 3: Arduino ESP8266 con Caudalímetro

Este caso es una ampliación del caso 2, el cual incluye un caudalímetro (YF-S201) conectado a una nueva bomba con su relé para regar otra maceta. La idea inicial era incluir más sensores como el de temperatura ambiente y otro sensor de humedad más para la maceta 2, pero el arduino ESP8266 tiene limitaciones en cuanto a entradas analógicas, sólo dispone de la A0. Se podría haber solucionado comprando un nuevo componente (ADC1115) para convertir las señales digitales en analógicas.

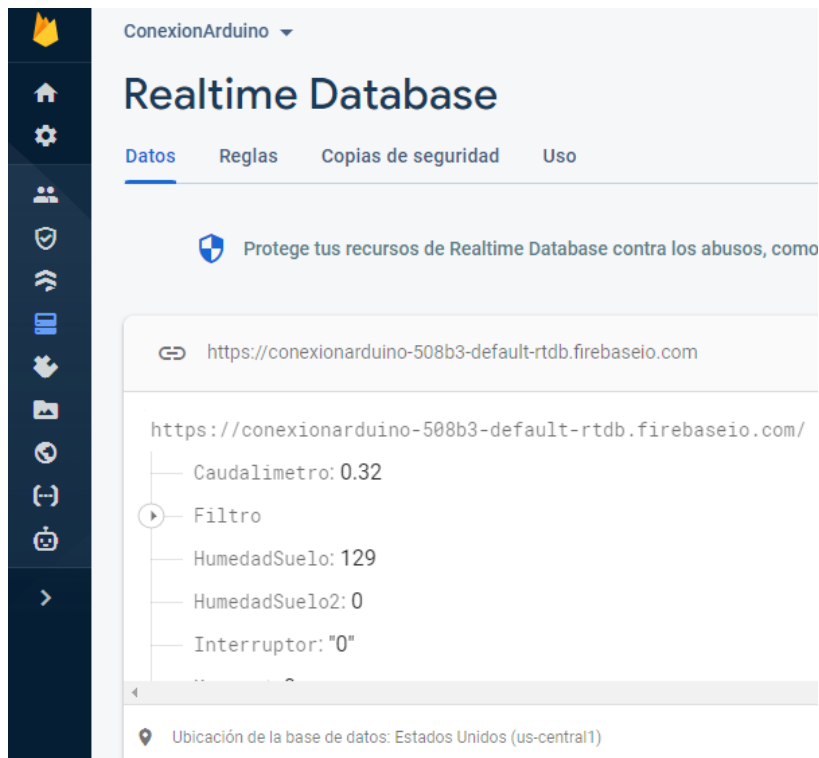


## 7. Firebase + AppInventor

He utilizado Firebase por ser una base de datos que pertenece a Google sin coste alguno con estructuras de datos json (clave / valor), la cual se puede integrar de forma fácil con el software de arduino y con AppInventor para desarrollar App para Smartphone.

La estructura de datos creada es la siguiente:





Pantallas de la App creada con AppInventor:



Fig 1



Fig.2



Fig.3



Fig.4

La Fig 1 es la pantalla que muestra la App cuando el Control de Riego se encuentra sin regar debido a que el valor del sensor de humedad del suelo se encuentra con un valor de humedad superior a 300, La Fig. 2 nos indica que ha saltado el riego automático; el valor del sensor es inferior a 300. La Fig. 3 hemos seleccionado de forma manual el botón de la maceta1 y se ha procedido al riego manual activándolo desde la App. La Fig 4 nos muestra las 2 macetas activadas de forma manual y regándose de forma simultanea.

Los datos se van actualizando en la pantalla del Smartphone en tiempo real recogiendo las actualizaciones realizadas por el arduino en la base de datos Firebase.

## 8.- Conclusiones

Se ha automatizado un Arduino para que riegue de forma automática una maceta cuando el sensor de humedad detecta que la tierra está seca, se ha creado una App para visualizar el estado de riego y además poder controlar el riego de forma manual las dos macetas.

Para un próximo trabajo añadiré los sensores de temperatura y de humedad para la maceta 2 con el convertidor ADC1115.

Se podría almacenar datos históricos por fechas y horas para poder obtener estadísticas.

Tenemos la opción de recoger los datos de la AEMET a través de una API con claves y predecir lluvias durante la siguiente semana (con Python se podría automatizar dichas consultas).

En la App se podría incluir filtros de horas, caudal, temperaturas, para poder realizar un riego programado en las macetas.

Con todas estas opciones conseguiremos un Control de Riego automatizado, programado con filtros e inteligente.

***Arduino + Firebase + AppInventor = Control de Riegos Inteligente***

## 7.- Bibliografía

<https://sites.google.com/view/controlriego/home>

<https://github.com/FirebaseExtended/firebase-arduino>

[https://www.youtube.com/watch?v=idJDYJ0PtWs&ab\\_channel=Alex7Tutoriales](https://www.youtube.com/watch?v=idJDYJ0PtWs&ab_channel=Alex7Tutoriales)

<https://www.arduino.cc/reference/es/language/structure/control-structure/dowhile/>

<https://forum.arduino.cc/t/problema-con-nodemcu-y-relays/531124/7>

<https://programarfacil.com/blog/arduino-blog/rele-con-arduino-lampara/>

<https://create.arduino.cc/projecthub/pulasthi-Narada/connecting-esp8266-to-firebase-to-send-receive-data-4adf66>