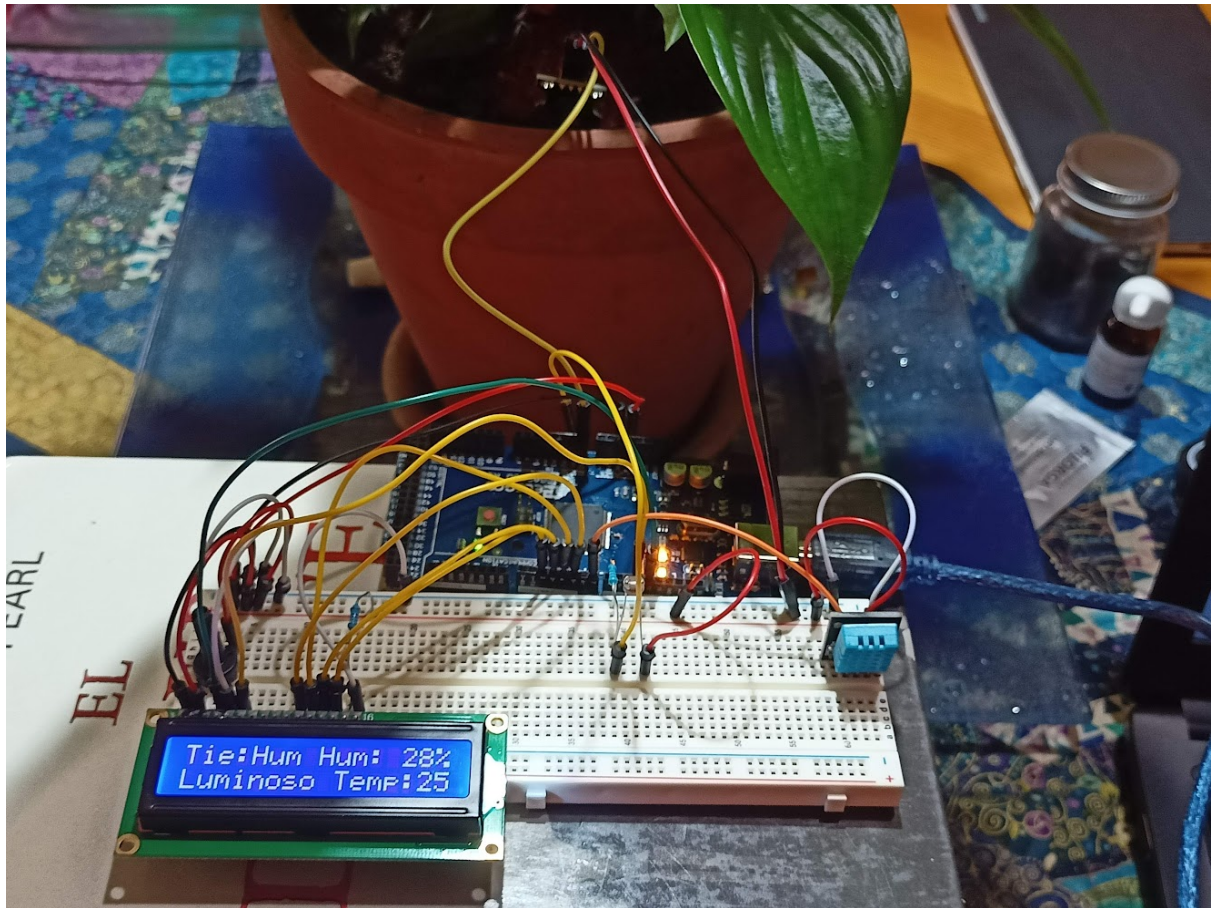


# Cuidar una planta con Arduino



## 1. Explicación del problema

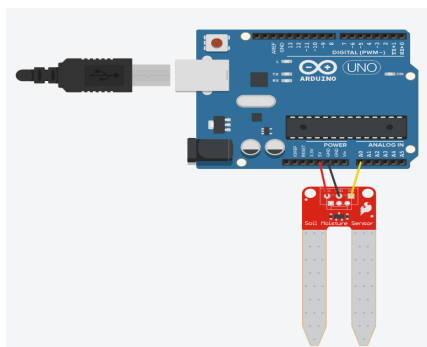
Las plantas son un buen método de decorar una casa y darle algo de vida a una estancia, pero no muchas personas son capaces de cuidar adecuadamente de ellas. En la mayoría de casos, el problema reside en saber cuándo hay que regar o donde es el lugar óptimo donde colocar la maceta. Con el uso de Arduino, he buscado resolver estas dudas de forma empírica.

La planta que tengo en mi casa se denomina Lirio de la Paz o Cuna de Moises por la forma de sus flores. Lo primero que hice fue saber qué era necesario para su cuidado óptimo:

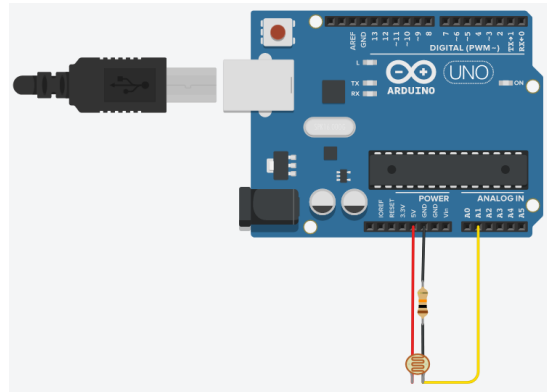
- Agua. En un principio, este no parecía ser uno de los problemas. Esta planta tiene la característica de que, cuando comienza a faltarle agua, sus hojas comienzan a decaer como si se desmayara. La cuestión es que no se debe regar por la parte de arriba. Para hacerlo correctamente, hay que poner la maceta en un recipiente con agua para que la absorba por las raíces. Por lo tanto, la pregunta era: ¿cuándo puedo quitar la maceta del agua? No queremos que la planta se ahogue, pero hay que dejar que la tierra se humedezca para que aguante bien el calor.
- Luz. Leyendo en internet supe que esta planta necesita unos niveles medios de iluminación. No puede darle la luz directamente del Sol porque sus hojas se queman, pero sí que tiene que estar en una estancia bien iluminada. Para ello, debía medir en qué habitaciones de la casa podría tener mejores condiciones.
- Temperatura. Por último, es una planta de clima no muy cálido. La temperatura para que pueda estar en buenas condiciones debe estar entre los 20 y 25 grados, con un margen de hasta 30 grados pero con mayor riego.

## 2. Partes del proyecto

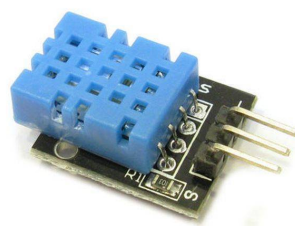
En primer lugar, para poder medir la humedad de la tierra en la parte alta de la maceta y así saber cuándo hay que quitar o poner el agua, hay que usar un sensor de humedad de la tierra. Su funcionamiento es sencillo, solo hay que conectar los cables según indica el propio elemento (S para la señal, G para tierra y V para el voltaje). En la parte de programación, solo debemos recoger los valores obtenidos por el sensor en una variable y trabajar con ellos como queramos.



Para poder medir la luz, hay que usar la fotorresistencia. Similar al funcionamiento del sensor de humedad de la tierra, conectamos una patilla al voltaje y otra a tierra con una resistencia de 10k y también a A1. De igual forma, solo debemos tomar los valores recogidos por el sensor en una variable para trabajar con ellos.



El último elemento de medición que he usado es el sensor de temperatura y humedad del aire. Este sensor no está incluido en Tinkercat y no tiene escritas las letras de a qué debe conectarse, pero tras una breve búsqueda en internet pude averiguar que se debe conectar de derecha a izquierda como tierra, potencia y salida. En mi caso, lo conecté al punto 7. Al igual que con los otros dos sensores, queremos tomar los datos en una variable, pero en este caso necesitamos el uso de la librería DHT11 para crear un objeto que permite hacer la lectura de la temperatura y la humedad.

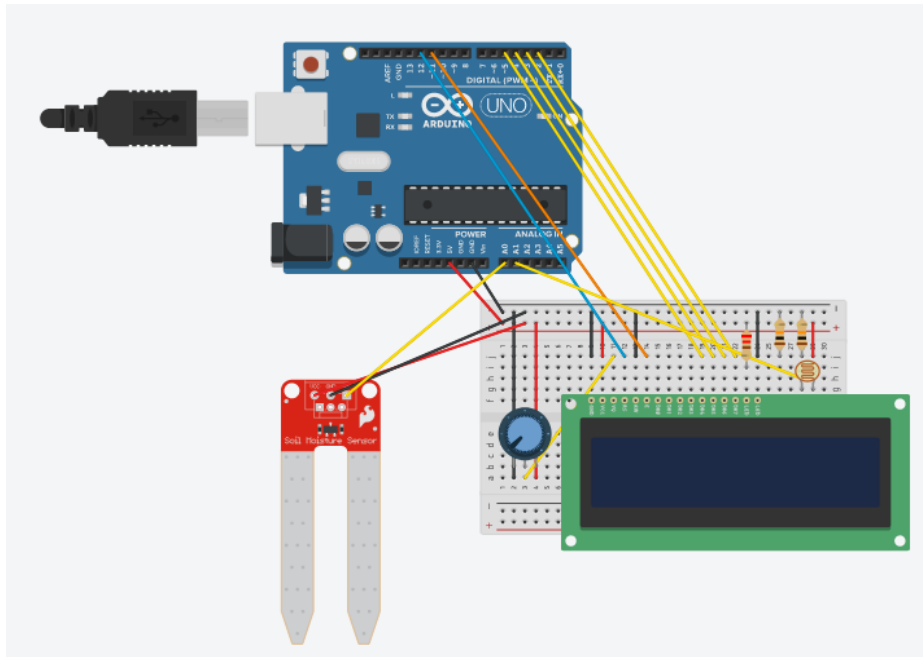




Una vez probado cada parte del proyecto, solo queda unirlo todo en una sola escena y programar la funcionalidad completa. Tras crear e inicializar todas las variables para cada sensor, comenzamos a trabajar con ellas. Tras las pruebas con el sensor de humedad de la tierra, calibré que el valor óptimo para la humedad requerida era por encima de 300. Por debajo de este valor, en la pantalla se dice que está seca y por encima que está húmeda. Para la humedad ambiental, con la librería obtenemos directamente el porcentaje así que solo hace falta mostrarlo tal cual por pantalla. Con la iluminación, ocurre lo mismo que con la humedad de la tierra. Como no necesita una gran cantidad de luz, con tener un valor de 300 como límite es suficiente. Por último, mostramos el valor de la temperatura tal cual.

Cabe puntualizar que, para poder escribir en dos filas distintas de la pantalla lcd, lo que he hecho es mover el cursor con la orden de la librería que usa la pantalla, que además he mostrado la información también por el monitor de arduino y que al final he puesto un delay de 5 segundos para que no se actualice constantemente.

Cabe puntualizar que, para poder escribir en dos filas distintas de la pantalla lcd, lo que he hecho es mover el cursor con la orden de la librería que usa la pantalla, que además he mostrado la información también por el monitor de arduino y que al final he puesto un delay de 5 segundos para que no se actualice constantemente.



*\*Falta el DHT11 que no está en Tinkercat\**

Código:

```
#include <LiquidCrystal.h> //librería para la pantalla LCD
#include "DHT.h" //librería para el sensor de temperatura

//inicialización de la pantalla LCD
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//inicialización del sensor de temperatura
#define DHTPIN 7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  lcd.begin(16, 2);
  dht.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  int humTierra = analogRead(A0);
  int luz = analogRead(A1);
```

```
lcd.clear();
int humAire = dht.readHumidity();
int tempC = dht.readTemperature();

//Humedad de la tierra
lcd.setCursor(0,0);
if (humTierra < 300)
    lcd.print("Tie:Seca ");
else
    lcd.print("Tie:Hum ");
Serial.print("Humedad tierra:");
Serial.print(humTierra);

//Humedad del ambiente
lcd.print("Hum: ");
lcd.print(humAire);
lcd.print("%");
Serial.print("|| Humedad aire:");
Serial.print(humAire);
Serial.print("%");

//Iluminación
lcd.setCursor(0,1);
if(luz<300)
    lcd.print("Oscuro");
else
    lcd.print("Luminoso");
Serial.print("|| Iluminacion:");
Serial.print(luz);

//Temperatura
lcd.print(" Temp:");
lcd.print(tempC);
Serial.print("|| Temperatura:");
Serial.print(tempC);
Serial.println("°C");

delay(5000);
}
```