

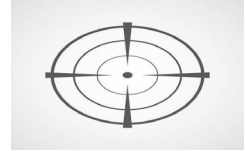
# **SERVO RIEGO AUXILIAR DE EMERGENCIA**



# **INDICE**

## **Tema 1**

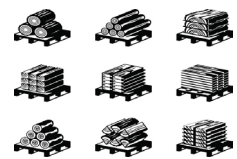
### **PROYECTO Y OBJETIVO**



- ⑩ **Página 1 – Proyecto y objetivo**
- ⑩ **Página 2 – Diagrama de bloques**

## **Tema 2**

### **MATERIALES**



- ⑩ **Página 3 – Servo para motor de riego**
- ⑩ **Página 4 – Sensor de humedad de tierra**
- ⑩ **Página 5 – Bomba de agua**
- ⑩ **Página 6 – Chip L293D y Relay**
- ⑩ **Página 7 – Protoboard**
- ⑩ **Página 8 – Caja para aislar el circuito y la Rasp.**

## **Tema 3**

### **CONEXIONES**



- ⑩ **Página 9 – Mapa de conexión bomb.Agua**
- ⑩ **Página 10 – Conexión servo aislado**
- ⑩ **Página 11 – Conexiones del sistema completo**

## **Tema 4**

### **PYTHON**



- ⑩ **Página 11 – Resumen**
- ⑩ **Página 12, 13 – Código completo**
- ⑩ **Página 14 – Código servomotor aislado**

## **Tema 5**

## **EFFECTIVIDAD**

- ⑩ **Página 15,16,17,18 Prueba 1**
- ⑩ **Página 18, 19, 20 Prueba 2**
- ⑩ **Página 21 Prueba 3**



## **Tema 6**

## **AUTOCRÍTICA**

- ⑩ **Página X Dificultades**
- ⑩ **Página X Fallos concretos**
- ⑩ **Agradecimientos**



# **Tema 1**

## **PROYECTO Y OBJETIVO**

### **Objetivo:**

La concepción de este riego es de emergencia, no tiene intención de sustituir un riego tradicional, está pensado para la persona que en su huerto no puede desplazarse a regar en algún momento.

Riego automatizado de emergencia;

Se activa el riego de 5 minutos cuando el sensor de humedad de la tierra detecta que hay poca humedad, Entoces iniciará un riego de 5 minutos, esperará 5 minutos para que le dé tiempo al agua a filtrarse en la tierra y volverá a tomar datos, si el porcentaje de humedad de la tierra es normal, el programa esperará 8 horas para volver a tomar datos, en caso de humedad baja, volverá hacer el ciclo del riego.

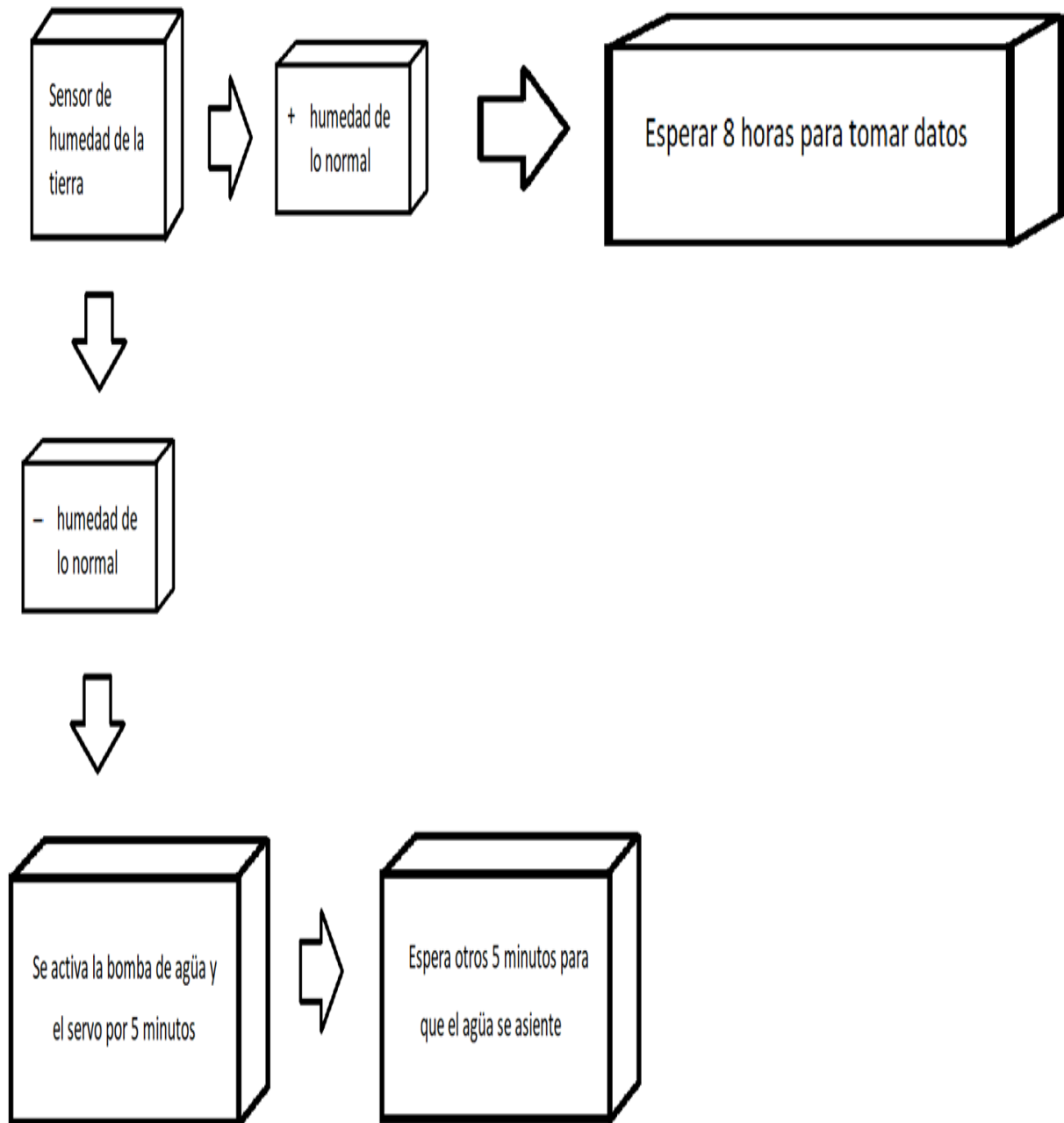
El funcionamiento se logra gracias a una Raspberry que controla todos los componentes; sensor de humedad de la tierra (modo digital) regulado por un potenciómetro, un chip L293D, un relay, una bomba de agua y un motor servo para regar

Este proceso es logrado gracias al código de Python desde la Raspberry, que controla en todo momento este sistema.

### **¡IMPORTANTE!**

Solo se deberá de ocupar la persona de rellenar el agua del bidón/botella/cubo.

# DIAGRAMA DE BLOQUES



## **Tema 2**

# **MATERIALES**

## **SERVO PARA MOTOR DE RIEGO**

### **Descripción:**

Zhiting 20kg, tensión de funcionamiento 4,8V y 7,2V 0,27A = 2Watts

### **Propiedades:**

Velocidad de funcionamiento: 0,17s/60grados(4,8v)

Velocidad de funcionamiento: 0,13s/60grados(6v)

Par de bloqueo: 12kg/cm(6v), 13kg/cm (7,2V)

Rotación de 180°, controlado por PWM

Cable rojo +, cable marron Ground y cable amarillo PWM

Precio: 14,07€ en Amazon.

Utilidades: Perfecto para regar pequeños huertos, jardines y plantas de tamaño medio, grande.

### **Experiencia personal:**



# SENSOR DE HUMEDAD DE TIERRA

## Descripción:

YL-69 Dolla Tek, tensión de funcionamiento: 3,3V y 5V, más información: [Cómo utilizar un sensor de humedad de suelo con Arduino \(automatizacionparatodos.com\)](http://Como utilizar un sensor de humedad de suelo con Arduino (automatizacionparatodos.com))

## Propiedades:

Modo de módulo de salida dual, salida digital(D0) no se necesita librería (en Digital se regula a mano a través de un potenciómetro, la salida analógica (A0) es más precisa.

**VALORES DEL SENSOR:** (0 máxima humedad - 1023 seco completamente)

Este modo **SOLO** funciona para el sensor para **ANALOGICO**



Precio:

Amazon

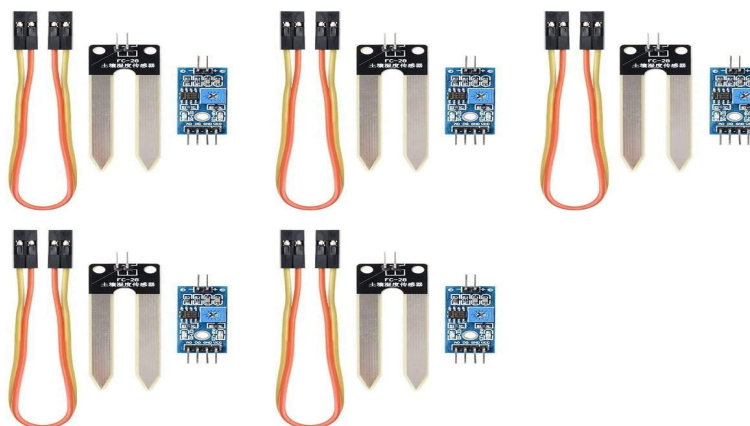
9,99€ en

## Utilidades:

Se trata de un sensor de agua simple que se puede utilizar para detectar la humedad del suelo.

Será útil para recordar que se debe regar las plantas de interior o para monitorear la humedad del suelo en su jardín o huerto.

**Experiencia personal:** A menor envío de señales, menor oxidación y mayor durabilidad.



# **BOMBA DE AGÜA**

**Descripción:** Runcci-Yun 3pcs Micro Bomba sumergible, tensión de funcionamiento 3v y 5v 100-200ma

**Propiedades:** Su gran caudal que puede ser 100L/H  
Servicio continuo de más de 300 horas

**+ 3 Metros de cable PVC con las siguientes características:**

- Ⓢ diámetro interior 5,54mm
- Ⓢ diámetro exterior 8,20mm
- Ⓢ superficie interior lisa para facilitar el flujo

**Precio:**  
11,99€ en Amazon

**Utilidades:**  
Es Ideal para acuarios, plantas de riego...

**Experiencia personal:**

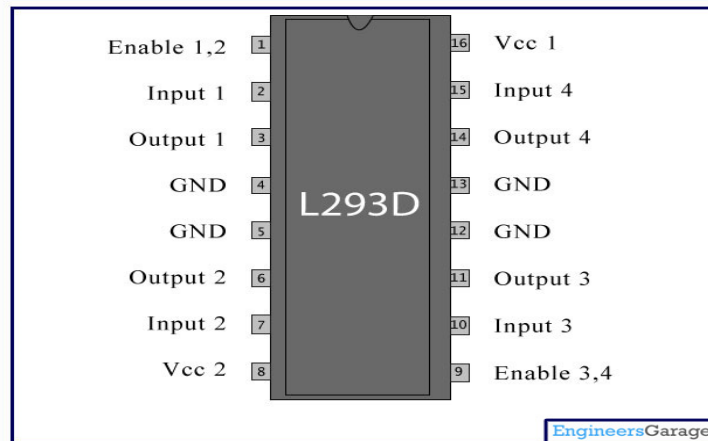




# CHIP L293D

**Descripción:** Chip controlador L293D

**Utilidades:** Permite múltiples conexiones y dispositivos, controla dispositivos mecánicos gracias a este chip y un relay.

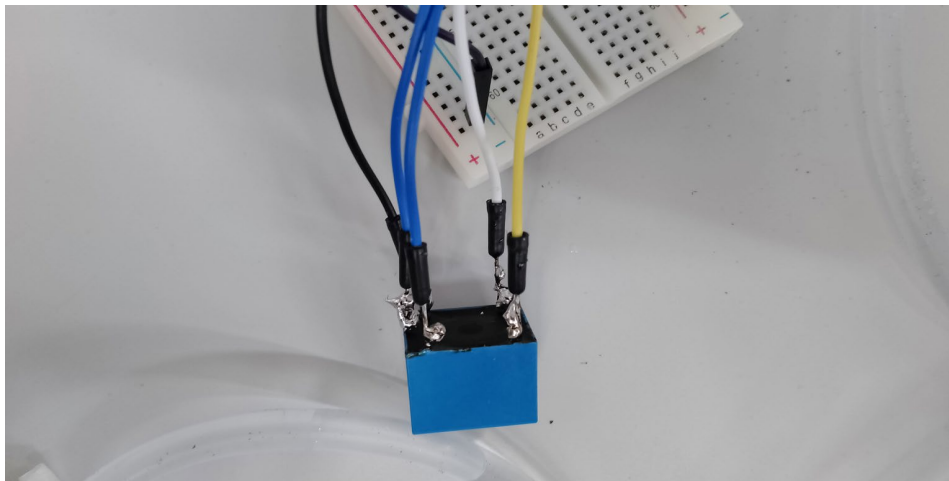


# RELAY

**Descripción:** permite abrir o cerrar el paso según se diga en el chip L2930 y a través de las Raspberry, incluye un diodo rectificador 1N4007.

Este relay es para la bomba de agua, ya que se tiene que controlar con el Chip y un relay porque carece de un cable de data.

¡importante! se ha tenido que soldar a un cable macho/macho cada pata del relay porque fallaba el contacto el relay directamente a la protoboard, esto nos ha salvado que el relay esté fallando cada poco.

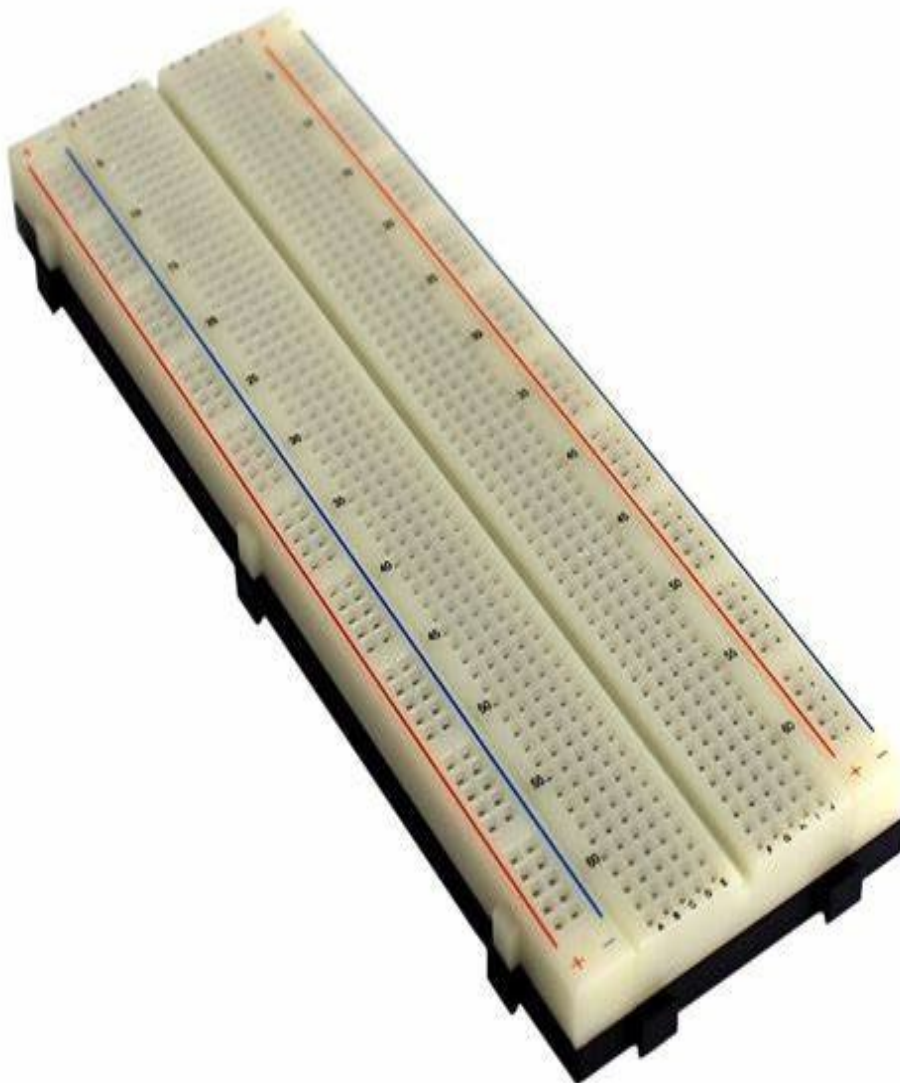


# **PROTOBOARD**

**Descripción:** Protoboard para las conexiones, para pruebas.

**Precio:** 5€ aproximadamente

**Utilidades:** montar todo el sistema eléctrico en ella.



# **CAJA PARA AISLAR EL CIRCUITO Y LA RASPBERRY**

**Materiales:** Cartón, cola blanca

**Herramientas:** Tijeras o cúter, regla y metro.

**Diseño:** en forma de barraca.

**Utilidad:** aislar el circuito y la raspberry del exterior.

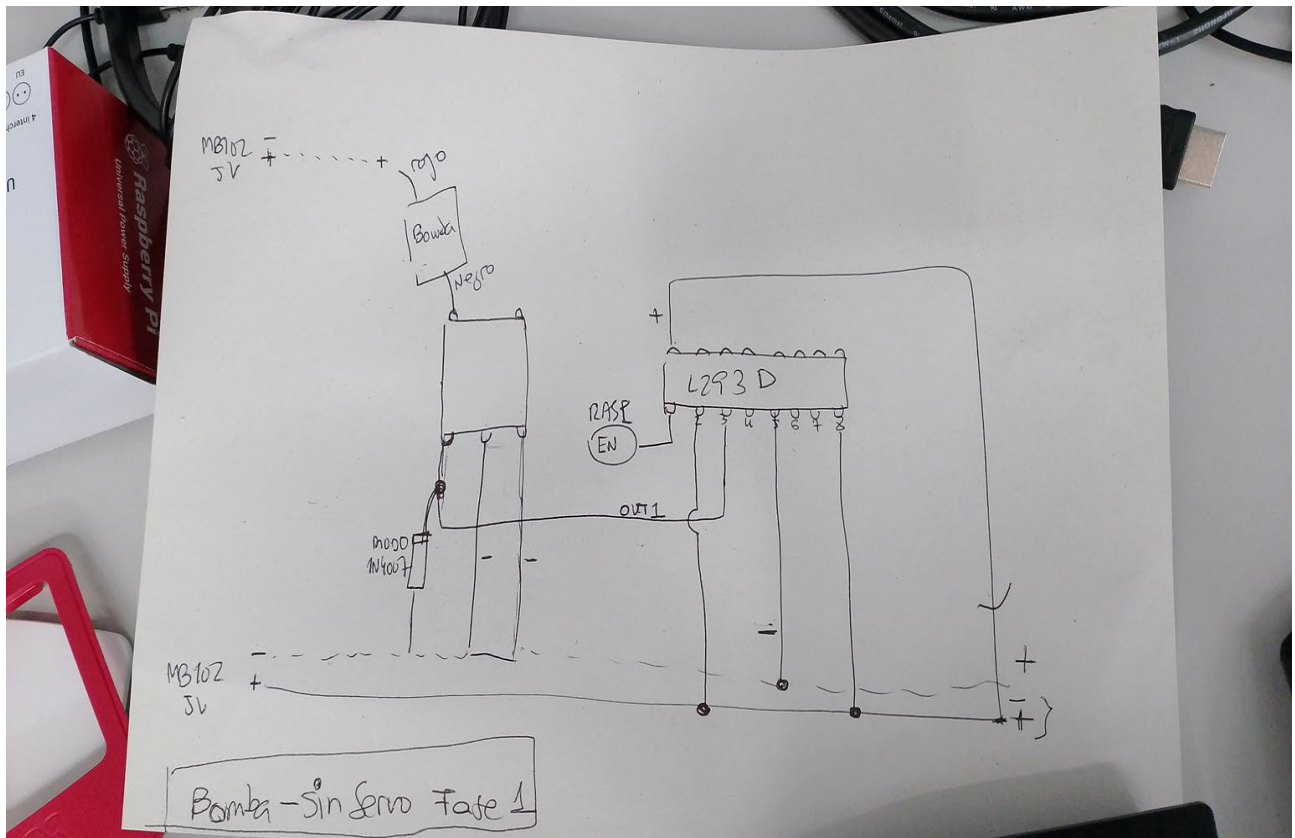


# TEMA 3

## CONEXIONES

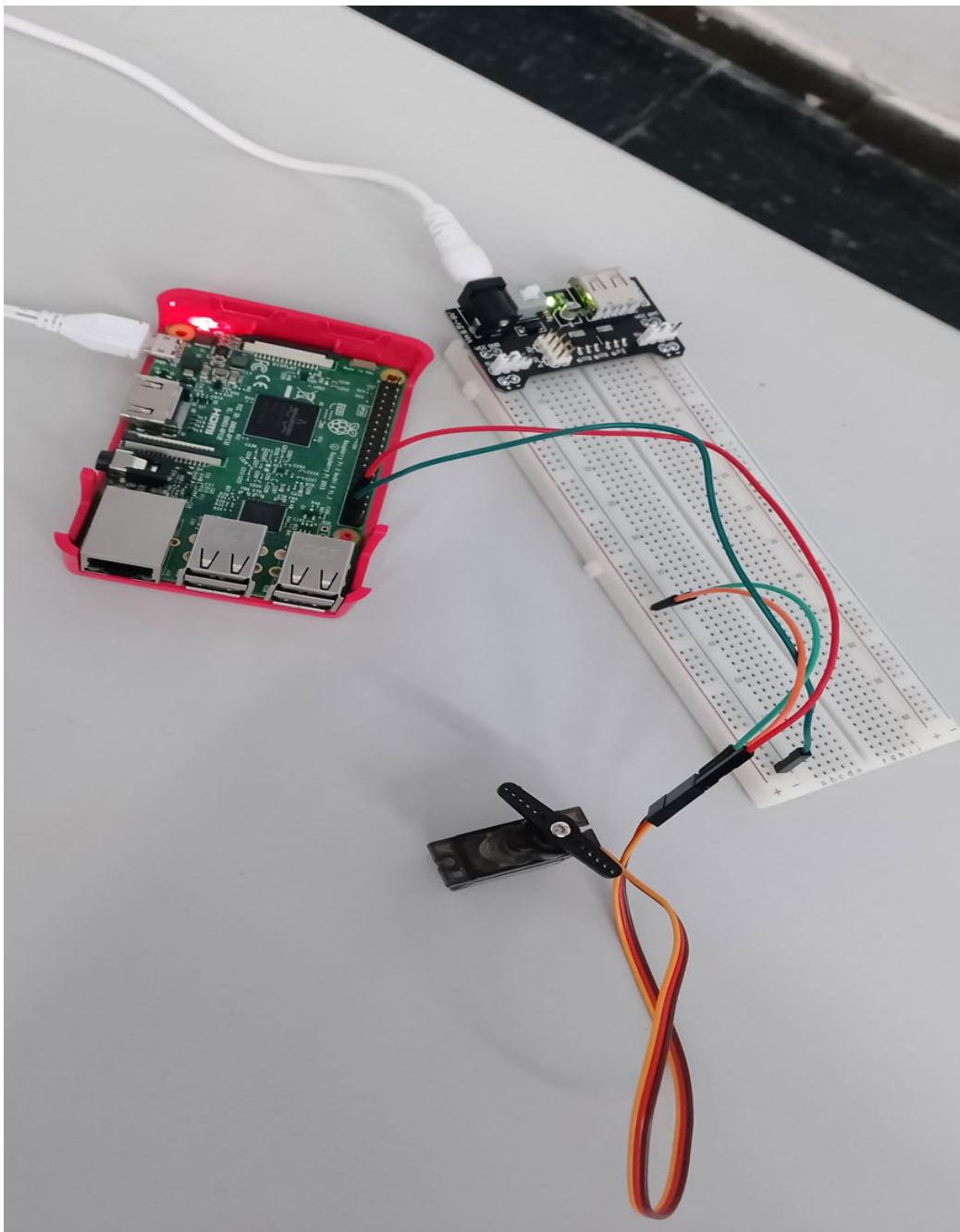


Conexión para la bomba de agua, se utiliza el chip L293D para controlar la bomba y un Relay, ver prueba1 en **TEMA5** EFECTIVIDAD 'Prueba 1'



Conexiones servo motor aislado, este código está en **TEMA 4 PYTHON** 'Código servomotor aislado', para ver la prueba del código ir a **TEMA 5 EFECTIVIDAD** 'Servomotor aislado'

Para el servo utilizamos un positivo con fuente externa 5V, un negativo, el PWM al pin 13 y un ground unificador a la Raspberry.

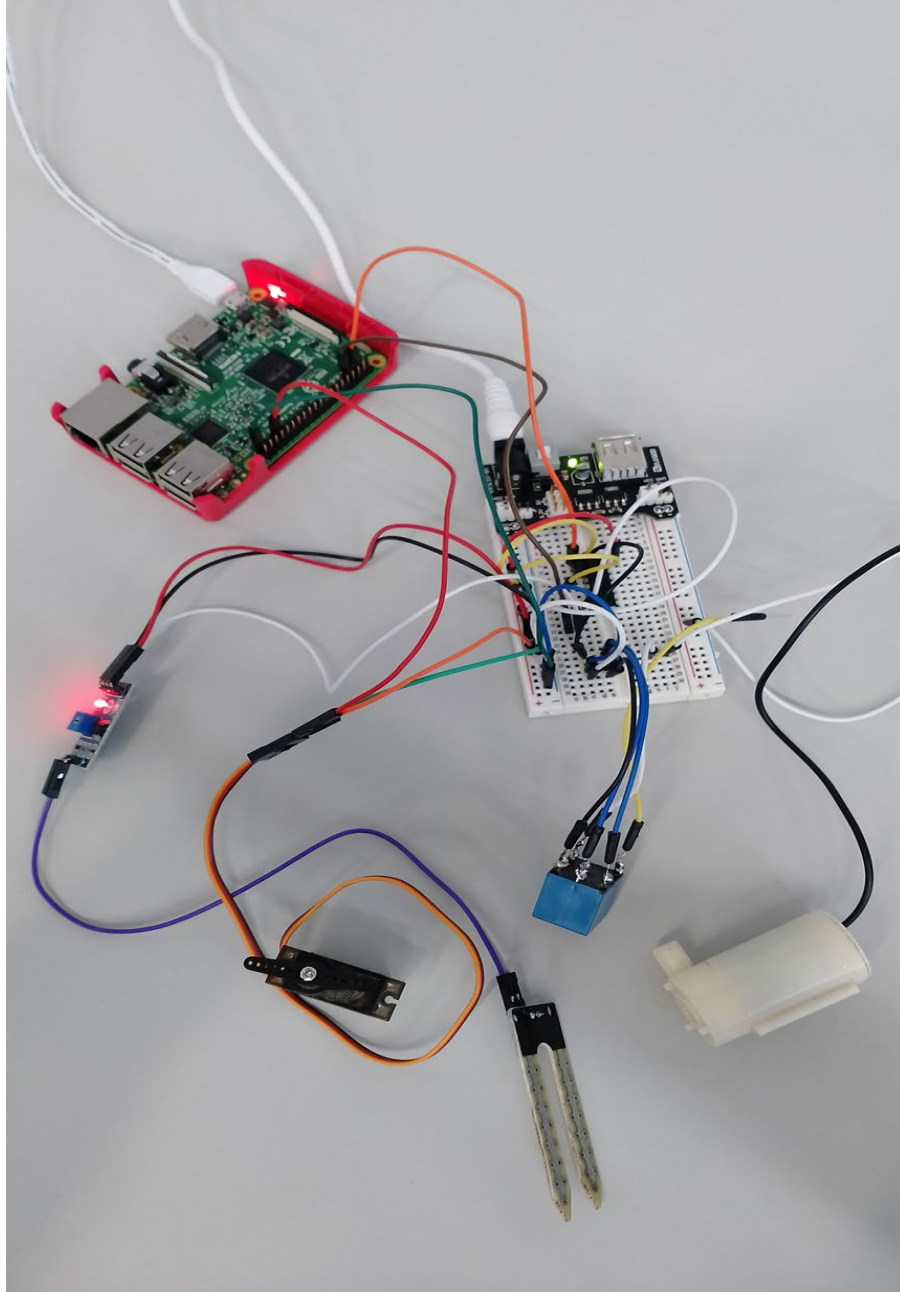


Conexiones completas L293D, sensor de humedad, relay, bomba de agua y



**servomotor.**

**La bomba de agua es la única conectada al L293D y al relay, los demás dispositivos se conectan en fuente de alimentación 5v positivo y ground + GPIO.**



## TEMA 4

# PYTHON



**Resumen;**

**El objetivo del código es;**  
al detectar una humedad inferior al propuesto manualmente en el sensor, (de forma digital) se activará el riego, cuando sea superior a la establecida manualmente no se activará.

**En todo momento nos avisará de la acción que va a tomar a través del print.**

**Para mas información sobre el objetivo de este proyecto ir al Tema 1 proyecto y objetivo.**

**El tiempo para tomar datos está especificado en el código, en las pruebas (TEMA5) se han reducido los tiempos de espera y de riego para comprobar su funcionamiento.**

**En las dos siguientes páginas, dejo el código de Python completo.**

# **CODIGO PYTHON**

## **(PONER FOTO DEL CONJUNTO y en efectividad la foto conjunta)**

```
from codecs import BOM
import RPi.GPIO as GPIO
import time
from re import T
import cayenne.client

# Credenciales de nuestro dispositivo Cayenne cliente y pines, llamar al cliente.
MQTT_USERNAME = "24e5acd0-4887-11ed-baf6-35fab7fd0ac8"
MQTT_PASSWORD = "a117cebb63dfddaec816128d8571693c0ef5bf95"
MQTT_CLIENT_ID = "2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6"
client = cayenne.client.CayenneMQTTClient()
SERVO= 2
BOMBA_DE_AGUA= 3
SENSOR_DE_HUMEDAD_TIERRA = 4

# Parametros de Cayenne y setup
def parametros (usuario, password, cliente_id):
    client.begin (usuario, password, cliente_id)

def cayene (valors):
    SENSOR_DE_HUMEDAD_TIERRA = valors[0]
    client.virtualWrite (1,SENSOR_DE_HUMEDAD_TIERRA,'hum')
    print('hum', SENSOR_DE_HUMEDAD_TIERRA)
    time.sleep(5)

def setup():
    GPIO.setmode(GPIO.BCM)
    parametros (MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID)
    GPIO.setup (SENSOR_DE_HUMEDAD_TIERRA, GPIO.IN)
    GPIO.setup ([SERVO, BOMBA_DE_AGUA], GPIO.OUT)
```



# Prepara el riego, la bomba se tiene que activar 3 segundos antes de poner el servo, todo en high.

def activarriego():

GPIO.output (BOMBA\_DE\_AGUA, GPIO.LOW)

print ('Cargando agua, espera del servo')

GPIO.output (BOMBA\_DE\_AGUA, GPIO.HIGH)

time.sleep(3)

GPIO.output (SERVO, GPIO.LOW)

print ('Servo activado')

GPIO.output (SERVO, GPIO.HIGH)

time.sleep(10)

GPIO.output (BOMBA\_DE\_AGUA, GPIO.LOW)

GPIO.output (SERVO, GPIO.LOW)

#Está en modo digital por lo tanto el resultado es 0 o 1

def regar():

if GPIO.input(SENSOR\_DE\_HUMEDAD\_TIERRA):

print ('activando el riego')

activarriego()

print ('riego terminado')

time.sleep(2)

print('preparandose para tomar datos, espere un momento')

time.sleep(3)

else:

print('No hace falta regar, humedad correcta')

time.sleep(2)

print('preparandose para tomar datos')

time.sleep(3)

def loop():

setup()

client.loop()

regar()

time.sleep(3)

if \_\_name\_\_ == '\_\_main\_\_':

while True:

loop()

# Código servomotor aislado (ir actualizando)

```
import RPi.GPIO as GPIO
import time

servoPIN = 13
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 13 for PWM with 50Hz
p.start(0)
try:
    while True:
        p.ChangeDutyCycle(5)
        print('movimiento 1')
        time.sleep(1)
        p.ChangeDutyCycle(7.5)
        print('movimiento 2')
        time.sleep(1)
        p.ChangeDutyCycle(10)
        print('movimiento 3')
        time.sleep(1)
        p.ChangeDutyCycle(12.5)
        print('movimiento 4')
        time.sleep(1)
        p.ChangeDutyCycle(10)
        print('movimiento 5')
        time.sleep(1)
        p.ChangeDutyCycle(7.5)
        print('movimiento 6')
        time.sleep(1)
        p.ChangeDutyCycle(5)
        print('movimiento 7')
        time.sleep(1)
        p.ChangeDutyCycle(2.5)
        print('movimiento 8')
        time.sleep(1)
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```

## Código completo threading

```
import threading
import RPi.GPIO as GPIO
import time
import queue

activaservo = queue.LifoQueue()
BOMBA_DE_AGUA= 3
SENSOR_DE_HUMEDAD_TIERRA = 4
SERVOPIN = 13
PAUSA = 2
PAUSA_CORTA= 0.5
RIEGO = 10
GPIO.setmode(GPIO.BCM)
GPIO.setup (SENSOR_DE_HUMEDAD_TIERRA, GPIO.IN)
GPIO.setup (BOMBA_DE_AGUA, GPIO.OUT)
GPIO.setup(SERVOPIN, GPIO.OUT)
p = GPIO.PWM (SERVOPIN, 50)
p.start(0)

def serving():
    activar = activaservo.get()
    if activar:
        p.ChangeDutyCycle(7.5)
        print('chorro central')
        time.sleep(2)
        p.ChangeDutyCycle(9)
        print('chorro izquierda')
        time.sleep(2)
        p.ChangeDutyCycle(7.5)
        print('chorro al centro')
        time.sleep(2)
        p.ChangeDutyCycle(6)
        print('chorro a la derecha')
        time.sleep(2)
```

else:

    p.stop()

def activarriego():

    GPIO.output (BOMBA\_DE\_AGUA, GPIO.LOW)

    print ('Cargando agua - el servo permanece en espera')

    time.sleep(12)

    GPIO.output (BOMBA\_DE\_AGUA, GPIO.HIGH)

    time.sleep(PAUSA\_CORTA)

    print ('Bomba y agua desactivados')

    time.sleep(PAUSA\_CORTA)

def bombing():

    while True:

        time.sleep(PAUSA\_CORTA)

        if GPIO.input(SENSOR\_DE\_HUMEDAD\_TIERRA):

            time.sleep(PAUSA)

            activaservo.put(1)

            activarriego()

            print ('riego terminado')

            time.sleep(PAUSA)

        else:

            activaservo.put(0)

if \_\_name\_\_ == '\_\_main\_\_':

    bomba = threading.Thread(target=bombing)

    servo = threading.Thread(target = serving)

    bomba.start()

    servo.start()

## TEMA 5

# EFFECTIVIDAD

Prueba 1

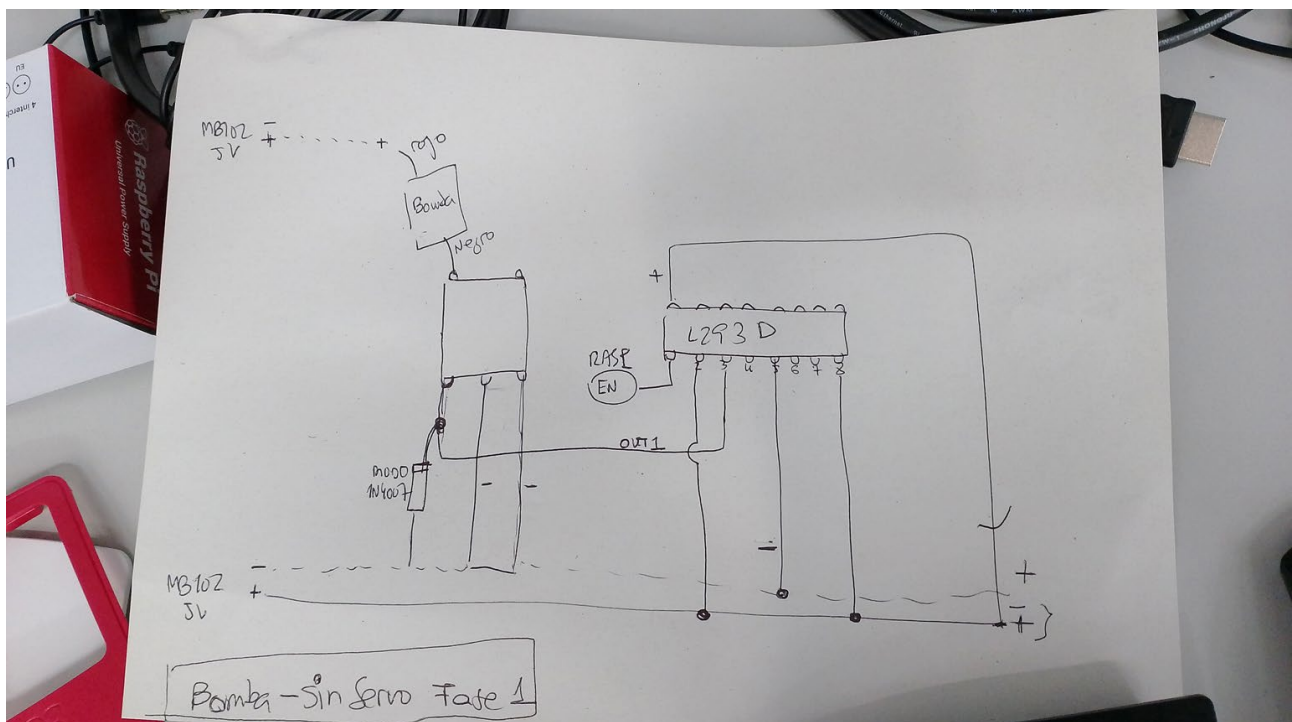
Fecha: 26/10/2022 12:30 aprox

## Descripción de la prueba

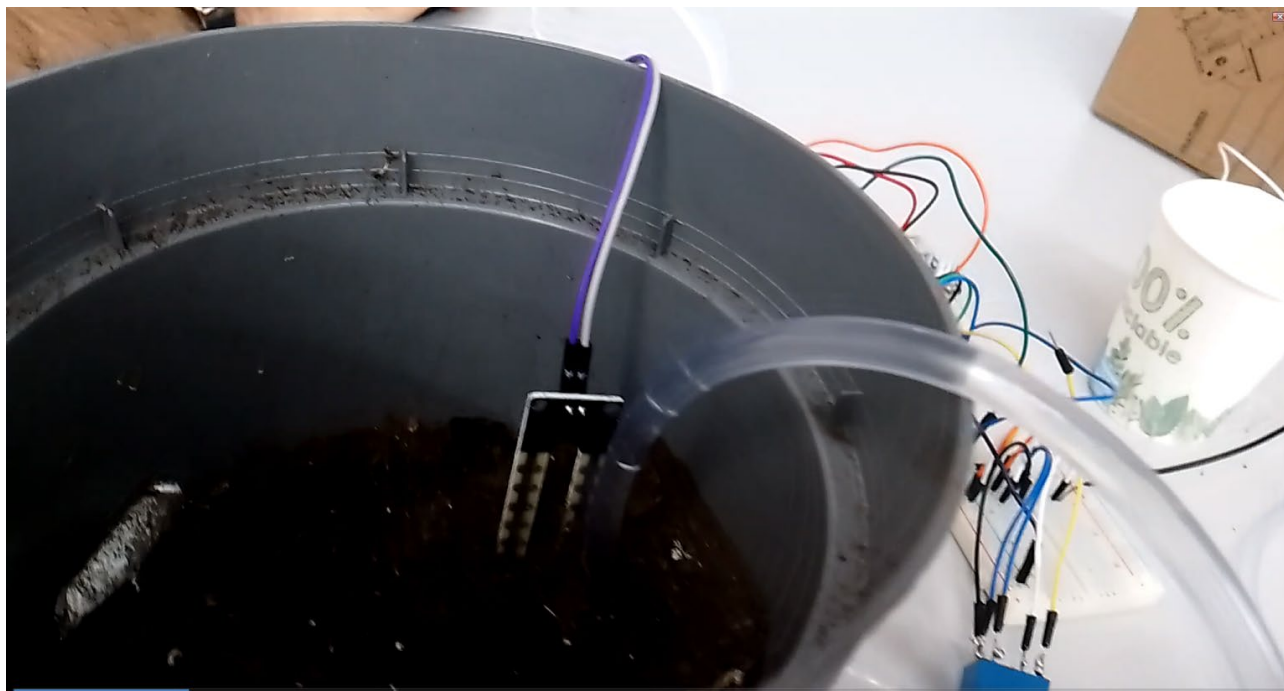
He activado el riego por 12 segundos solo con la bomba de agua GPIO3 y con el sensor de humedad GPIO4, sin el servo motor, al finalizar el riego 2 segundos más para volver a tomar datos, una vez que toma datos y nota que la humedad es suficiente ya no regará. En caso de regar como no regar, nos informará a través del print.

¡Importante! Tener mucho cuidado con el agua y la electricidad, es bastante peligroso, asegurar cada movimiento al montar, una vez montado y también en funcionamiento, asegurar con papel alrededor de los envases, cubos o botellas a utilizar y por debajo añadir papel, poner todo envase con líquido en un lugar, que en caso de vuelco no dañe nada alrededor de él.

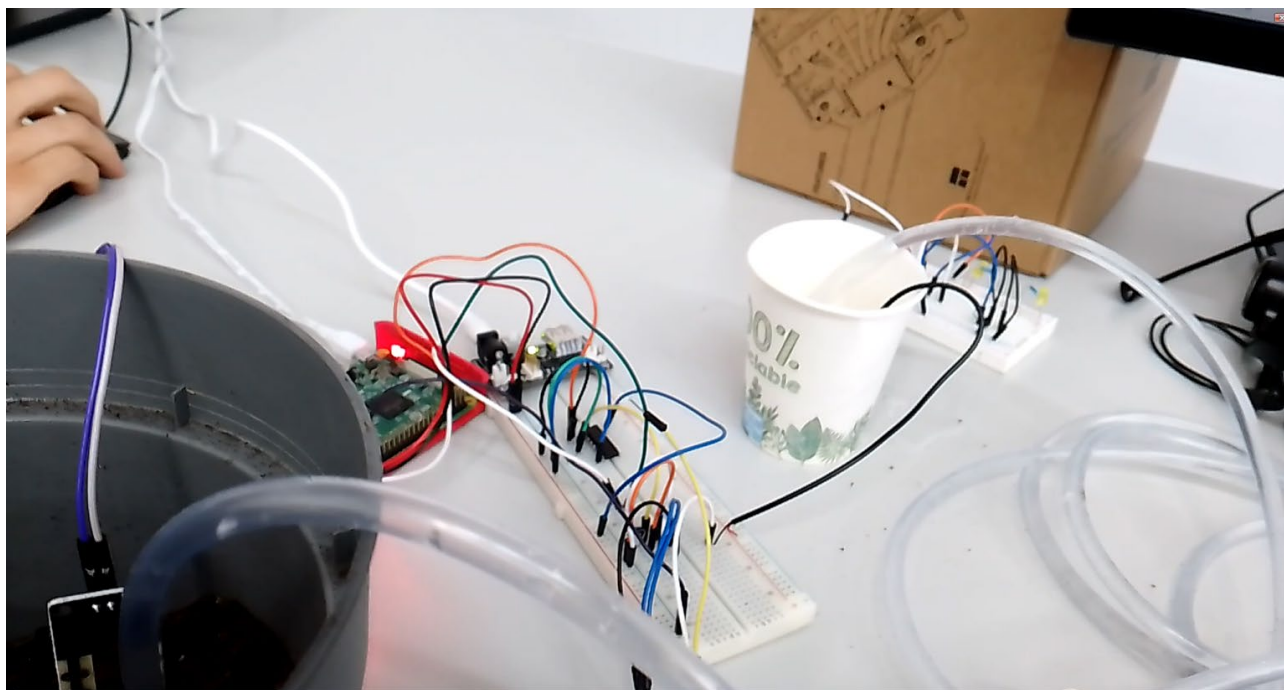
Conexiones relay + sensor +chip L293D sin servo + bomba de agua, conexiones:



Una vez montado toda la conexión eléctrica, el tubo de pvc de salida de la bomba a la maceta y el sensor de humedad en la tierra



La bomba de agua en su vaso con agua para la prueba, todas las conexiones eléctricas preparadas para ejecutar el código.



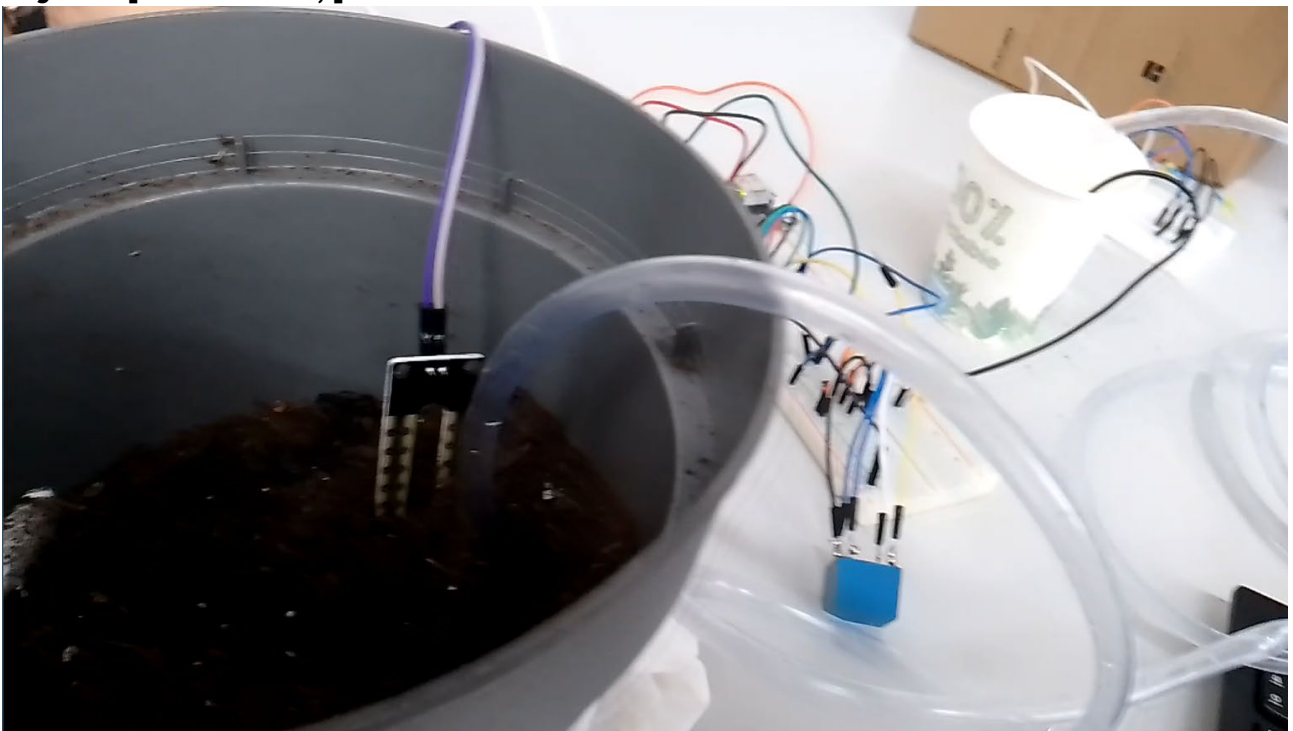
## Activamos el código

```
pi@raspberrypi: ~/Documents/ultimos/proyecto
time.sleep(5)
KeyboardInterrupt

pi@raspberrypi:~/Documents/ultimos/proyecto$ sudo python3 Proyecto_riego.py
Connecting to mqtt.mydevices.com:1883
/home/pi/Documents/ultimos/proyecto/Proyecto_riego.py:30: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup([SERVO, BOMBA_DE_AGUA], GPIO.OUT)
Connected with result code 0
SUB v1/24e5acd0-4887-11ed-baf6-35fab7fd0ac8/things/2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6/cmd/+
PUB v1/24e5acd0-4887-11ed-baf6-35fab7fd0ac8/things/2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6/sys/model
Python
PUB v1/24e5acd0-4887-11ed-baf6-35fab7fd0ac8/things/2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6/sys/version
1.1.0

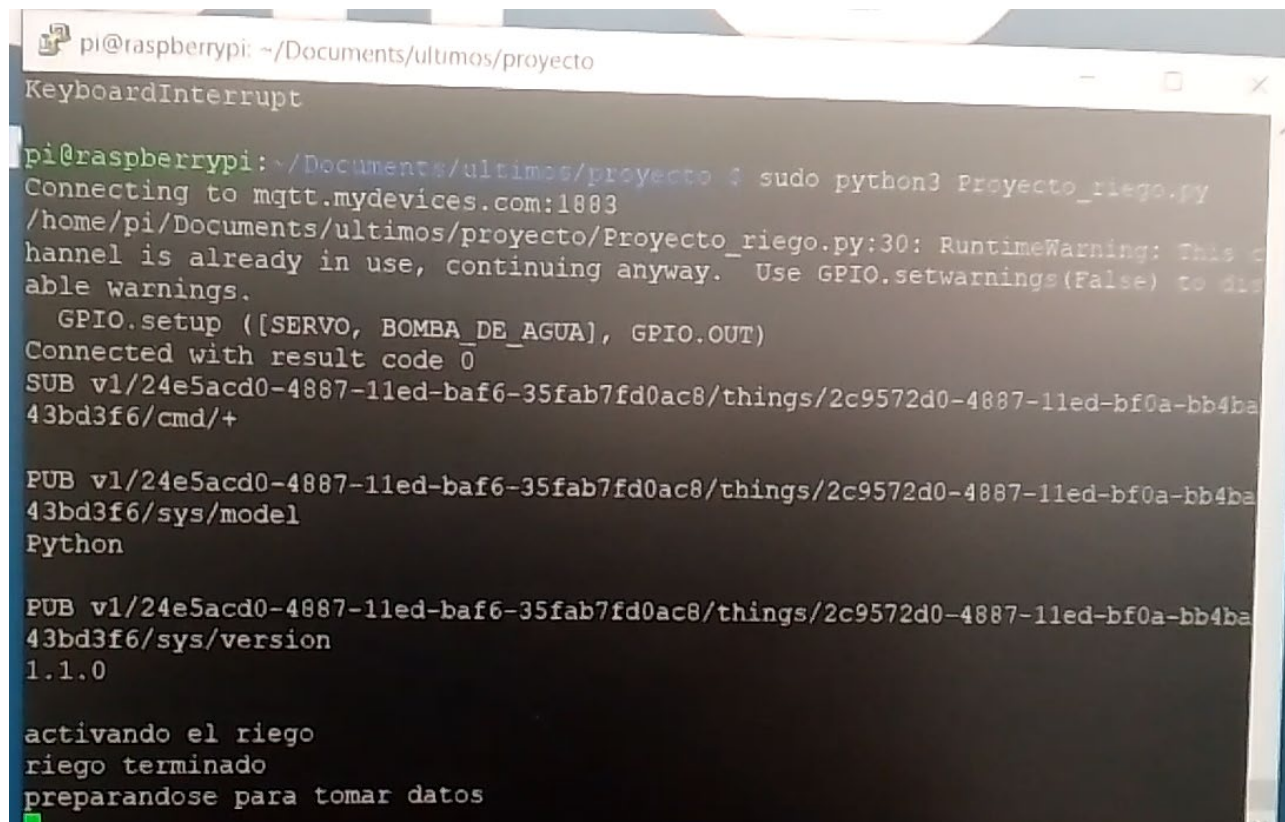
activando el riego
```

## Regando perfectamente, prueba exitosa





Todo perfecto, en la próxima prueba se le tendrá que aplicar el servo.



```
pi@raspberrypi: ~/Documents/ultimos/proyecto
KeyboardInterrupt

pi@raspberrypi:~/Documents/ultimos/proyecto $ sudo python3 Proyecto_riego.py
Connecting to mqtt.mydevices.com:1883
/home/pi/Documents/ultimos/proyecto/Proyecto_riego.py:30: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup ([SERVO, BOMBA_DE_AGUA], GPIO.OUT)
Connected with result code 0
SUB v1/24e5acd0-4887-11ed-baf6-35fab7fd0ac8/things/2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6/cmd/+

PUB v1/24e5acd0-4887-11ed-baf6-35fab7fd0ac8/things/2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6/sys/model
Python

PUB v1/24e5acd0-4887-11ed-baf6-35fab7fd0ac8/things/2c9572d0-4887-11ed-bf0a-bb4ba43bd3f6/sys/version
1.1.0

activando el riego
riego terminado
preparandose para tomar datos
```

## Prueba 2

4/11/2022 10:00 aprox

# Servomotor aislado

El código del servomotor aislado, está en **TEMA 4** (PYTHON).

El servo se ha probado de manera aislada en el pin 13, dejo fotos de la comprobación.

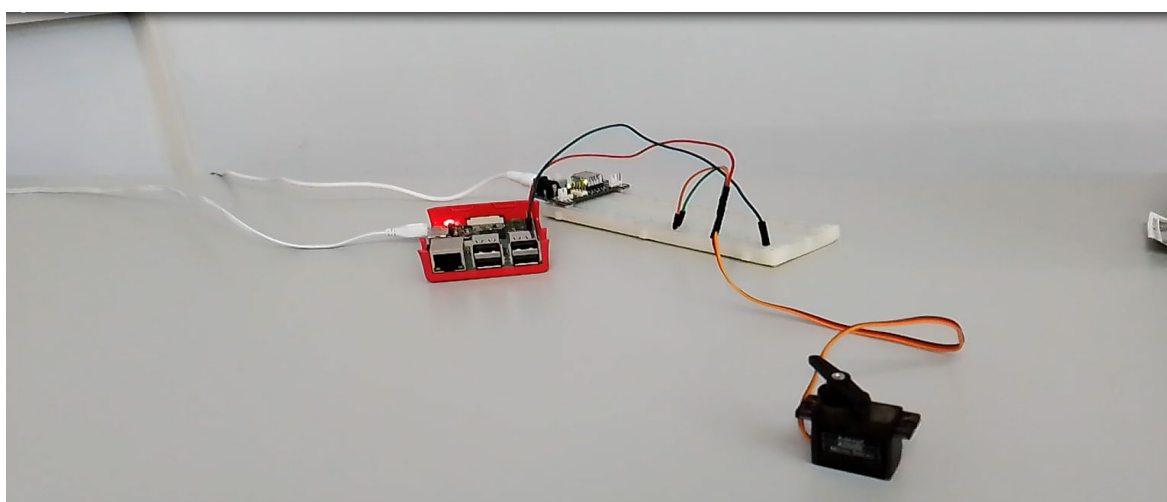
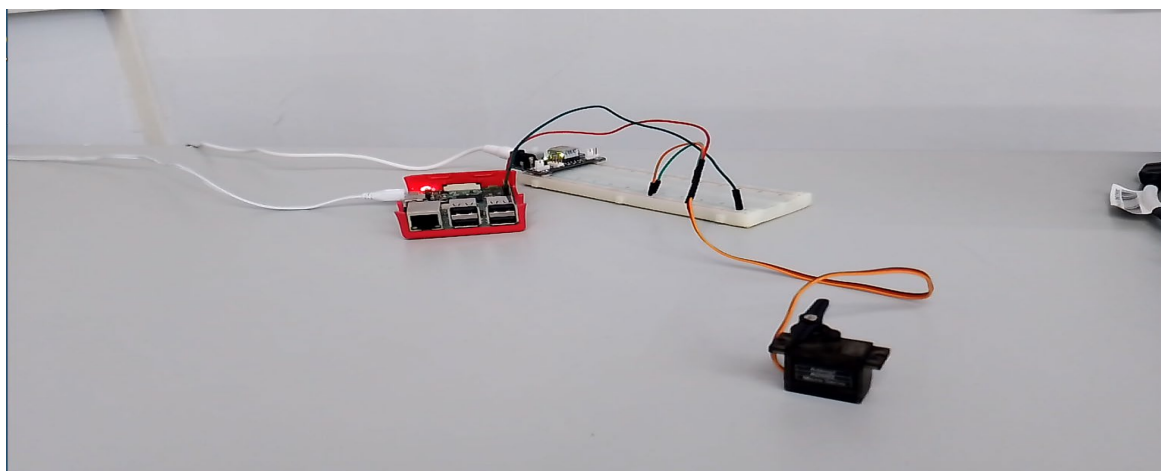
He tenido problemas en los servomotores, ya que tenían un defecto y no funcionaban correctamente, se ha tenido que probar en un servo de potencia inferior que funcionaba correctamente.

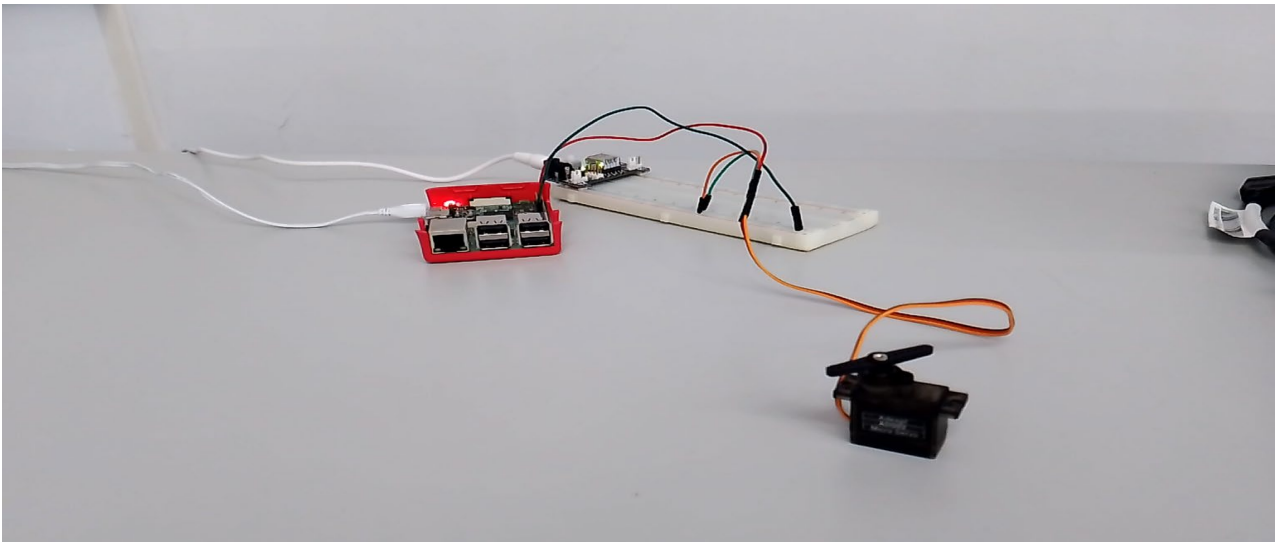
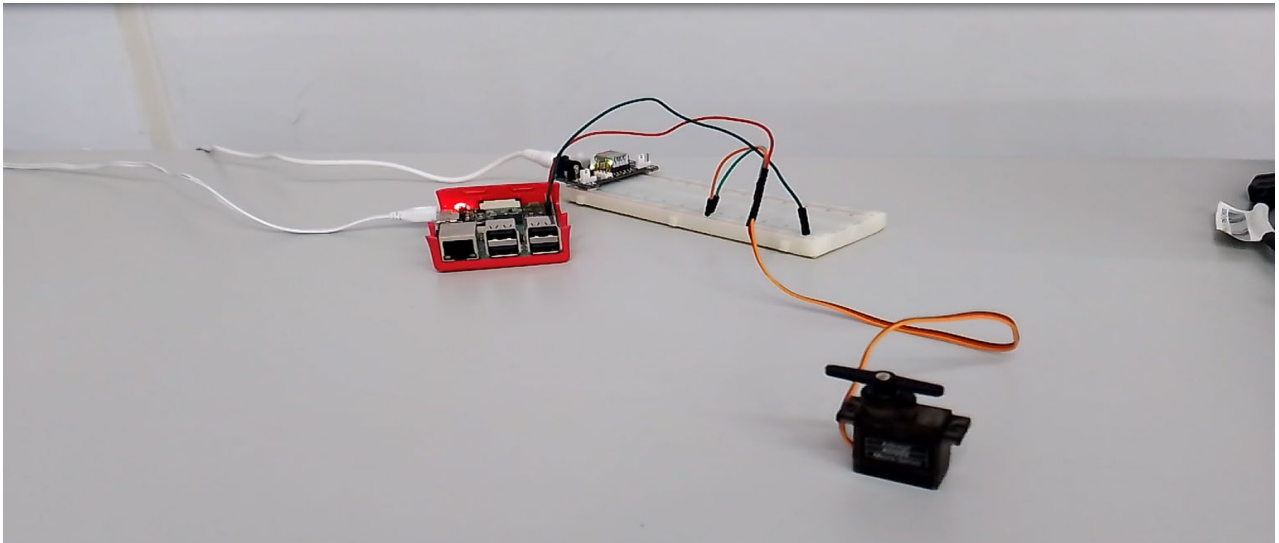
Impresion de los movimientos y el movimiento en fotos.



```
^Cpi@raspberrypi:~/Documents/ultimos/proyecto $ sudo python3 servo.py
movimiento 1
movimiento 2
movimiento 3
movimiento 4
movimiento 5
movimiento 6
movimiento 7
movimiento 8
movimiento 1
movimiento 2
movimiento 3
movimiento 4
movimiento 5
movimiento 6
^Cpi@raspberrypi:~/Documents/ultimos/proyecto $
```

A continuación algunas fotos que se corresponde, a los movimientos. Fijarse en el servo.



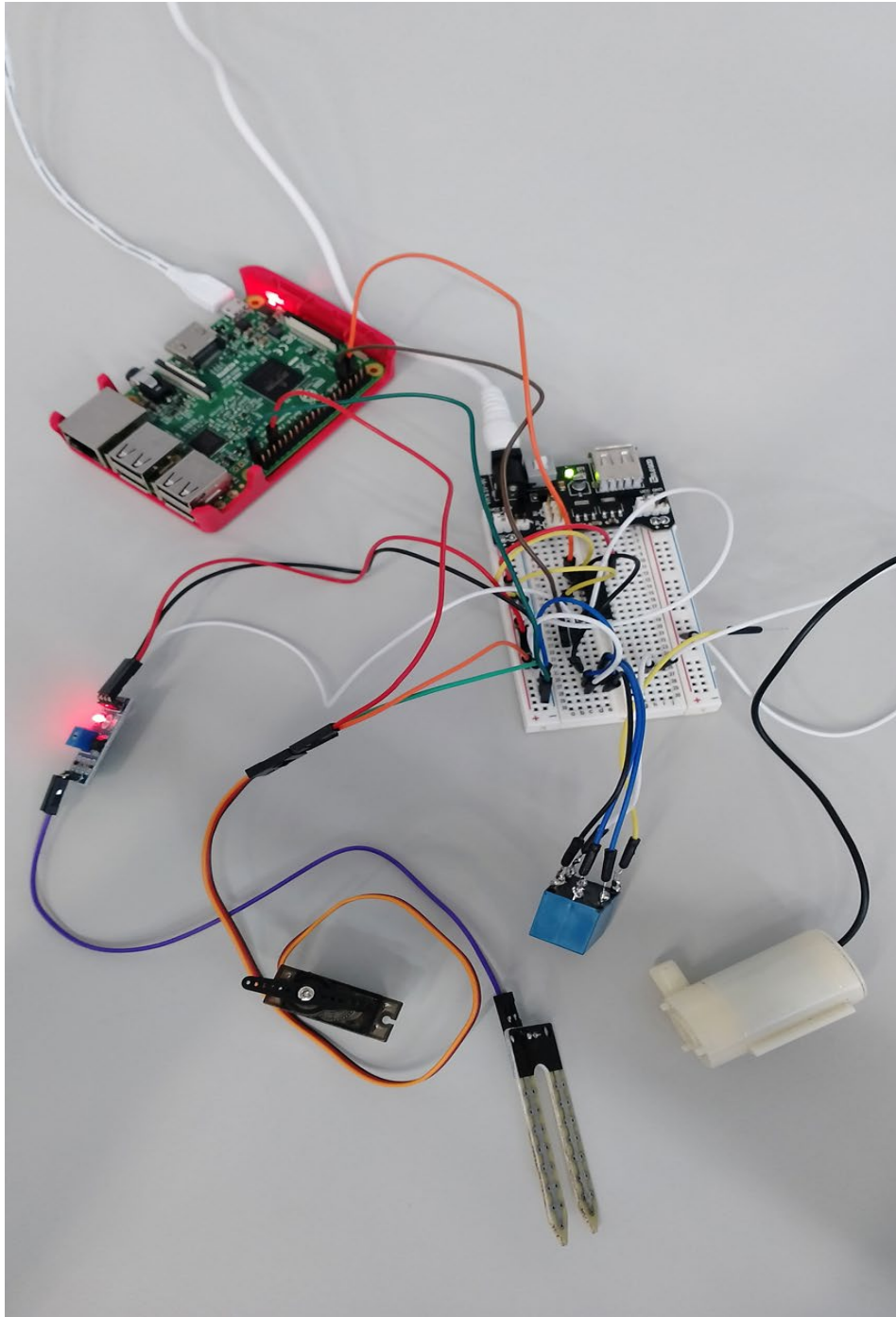


Prueba 3

4/11/2022 12:44 aprox

## **Sensor + bomba de agua + servomotor**

El código conjunto a fallado en el orden de ejecución del servo y el modo de funcionamiento del servo, la bomba de agua y el sensor funcionan perfectamente. Se tiene que integrar correctamente el código.



Prueba 4

8/11/2022 12:30 aprox

## **Fabricar la caja para aislar la Raspberry y el circuito**

Fabricamos una caja en forma de barraca para tener los componentes eléctricos aislados y centralizados, el tubo de la bomba el servo permanecen en el exterior, por la chimenea de la barraca.

Material base es el cartón, también se utiliza corcho para que la estructura sea mas resistente



**Procedemos a montar la caja**

**Las dimensiones son 31cm ancho x 20cm de alto y 20cm de profundo, utilizamos el corcho restante para apuntalar la barraca y que se pegue adecuadamente la cola blanca.**



**Por último el techo, debe hacer de tapa para poder sacar y meter la Raspberry y los componentes eléctricos.**

**El tubo y el servo pasará por la chimenea de la caja.**

## TEMA 6

# AUTOCRITICA

**Me ha dado mucha ilusión poder haber desarrollado este proyecto, he tenido muchas dificultades, empezar aprender Python desde cero es duro y traducir bastantes cosas del Ingles, ya que mi nivel es pésimo y debo ponerme aprender este idioma.**

**Pese a todo, he podido aprender muchísimo en el curso de IOT, mas de lo que imaginaba a pesar de no entender muchas cosas, he tenido que poner un esfuerzo extra de concentración y paciencia día a día para compensar mis desventajas, que no son pocas, y tener la humildad necesaria, que es el único camino para aprender y progresar en la vida, sea uno avanzado o novato.**

**Tengo claro que seguiré aprendiendo mas Python, mas Raspberry, Arduino y todo lo possible sobre el mundo de la informática.**

**Gracias al profesor Joan y su paciencia conmigo y gracias a todos los compañeros por este buen ambiente.**

