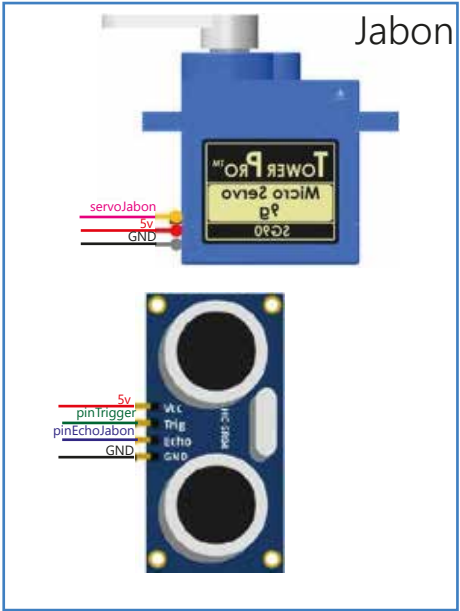
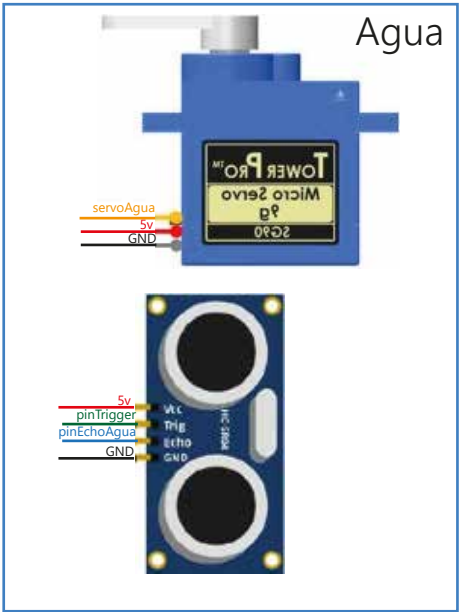
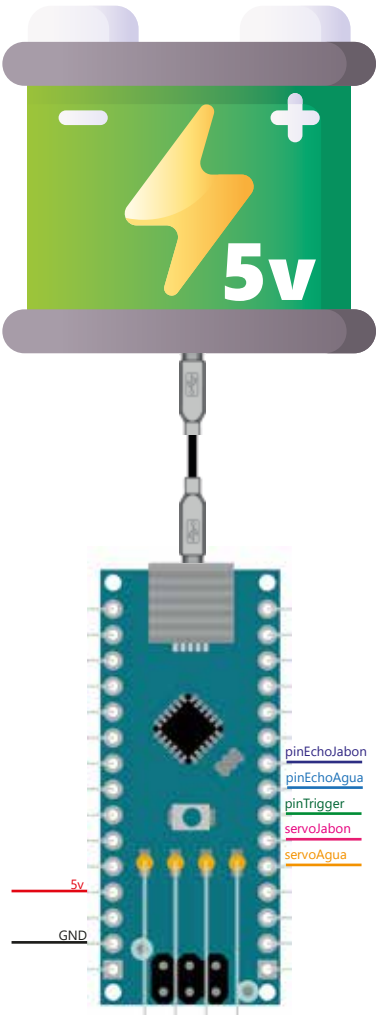


# Dispensador de Agua y Jabon con 2 Servos y 2 Sensores



```
// Dispensador con un servo y un rele
#define sbi(x, y) x |= _BV(y) // set bit
#define cbi(x, y) x &= ~(_BV(y)) // clear bit
#define tbi(x, y) x ^= _BV(y) // toggle bit
#define is_high(x, y) ((x & _BV(y)) == _BV(y))

// Configuramos los pines del sensor Trigger y Echo
#define pinEchoJabon 6 // señal de pinEchoAgua a pin 5
#define pinEchoAgua 5 // señal de pinEchoAgua a pin 5
#define pinTrigger 4 // señal de pinTrigger a pin 6

// Incluimos la librería para poder controlar el servo
#include <Servo.h>

// Declaramos la variable para controlar el servo
Servo servoAgua; // Servo para controlar el paso del agua
Servo servoJabon; // Servo para controlar el paso del jabón

unsigned int contador = 0; //valor del timer
float distancia;

void write_tmr1(unsigned int i)
{
    unsigned char sreg;

    //Guarda registro de estado
    sreg = SREG;
    //desabilita las interrupciones
    noInterrupts();
    //Carga el valor al timer1
    TCNT1 = i;
    //recupera registro de estado
    SREG = sreg;
}

unsigned int read_tmr1()
{
    unsigned char sreg;
    unsigned int i;
    //Guarda registro de estado
    sreg = SREG;
    //desabilita las interrupciones
    noInterrupts();
    //lee el timer1
    i = TCNT1;
    //recupera registro de estado
    SREG = sreg;
    return i;
}

void setup()
{
    Serial.begin(9600); // Iniciamos el monitor serie para mostrar el resultado

    pinMode(pinEchoAgua, INPUT);
    pinMode(pinEchoJabon, INPUT);
    pinMode(pinTrigger, OUTPUT);
    digitalWrite(pinTrigger, LOW); //para generar un pulso limpio ponemos a LOW 4us

    //Configura registroA
    TCCR1A = 0;
    //configura registroB
    TCCR1B = 0;
    TCCR1B = 0b010; // Configura el prescaler a 8
    write_tmr1(0);
    sbi(TIFR1, TOV1); // pone un uno para limpiar bandera de desborde

    // Iniciamos los servos
    servoJabon.attach(3);
    servoAgua.attach(2);
}

//entrega el pulso al Sensor ultrasonico y lee la distancia
float lee_distancia(char pin) //lee en el pin especificado
{
    float calculo;
    digitalWrite(pinTrigger, HIGH);
    delayMicroseconds(10);
    //limpia la bandera de desborde
    digitalWrite(pinTrigger, LOW);

    sbi(TIFR1, TOV1); // pone un uno para limpiar bandera de desborde

    while (digitalRead(pin) == 0) //espera que el eco se levante
    {
        if (is_high(TIFR1, TOV1))
        {
            sbi(TIFR1, TOV1); // pone un uno para limpiar bandera de desborde
            return (-1); // si se desborda el timer -1 sensor desconectado
            break;
        }
    }

    write_tmr1(0);

    while (digitalRead(pin) == 1) // hasta que caiga el eco
    {
        contador = read_tmr1(); //captura el valor del contador

        //si se desborda el timer activa la bandera de desborde, no hay obstaculos frente
        if (is_high(TIFR1, TOV1))
        {
            sbi(TIFR1, TOV1); // pone un uno para limpiar bandera de desborde
            return (0); // si se desborda el timer no hay obstaculos
            break;
        }
    }

    calculo = (float)contador * 0.0343 * 8.0 / (2.0 * 16.0); //en centimetro //prescaler=8 critical=16
    return (calculo);
}

void loop()
{
    distancia = lee_distancia(pinEchoJabon);
    if (distancia <= 15.0)
    {
        Serial.println("Dispensando Jabon");
        servoJabon.write(180); // Dispensamos jabon
    }
    else
    {
        servoJabon.write(0); // No Dispensamos jabon
    }
    Serial.print("d Agua: ");
    Serial.print(distancia); //Enviamos serialmente el valor de la distancia
    Serial.println("cm");
    delay(50);

    distancia = lee_distancia(pinEchoAgua);
    if (distancia <= 15.0)
    {
        Serial.println("Dispensando Agua");
        servoAgua.write(180); // Dispensamos agua
    }
    else
    {
        servoAgua.write(0); // No Dispensamos agua
    }
    delay(50);
    Serial.print("d Jabon: ");
    Serial.print(distancia); //Enviamos serialmente el valor de la distancia
    Serial.println("cm");

    delay(1000); // Hacemos una pausa de 500ms
}
```