

```

#include <LiquidCrystal_I2C.h> //Pantalla lcd
LiquidCrystal_I2C lcd(0x3F, 20, 4); //Interface I2C para la lcd, el 0x3F
es el modelo de la misma interfaz y la cual se obtiene desde la misma
interface con condigo
#include <EEPROM.h>          // libreria EEPROM

// Declaraciones para los ultrasonidos

int pin_TRIG = 6; //Pin para enviar pulsos (ultrasonido) tinaco 1
int pin_ECHO = 5; //Pin para recibir la señal de rebote (ultrasonido)
tinaco 1
int t2pin_TRIG = 4; //Pin para enviar pulsos (ultrasonido)tinaco 2
int t2pin_ECHO = 3; //Pin para recibir la señal de rebote (ultrasonido)
tinaco 2

//variables para las mediciones del ultrasonido del sensor del tinaco 1

int tiempoSenal; // tiempo de respuesta de recepcion de señal del módulo
ultrasonido
int litros; // variable para almacenar la converscion de volumen a litros
float distanciaM; //variable que almacena el valo de la distancia
considerada como altura

//variables para las mediciones del ultrasonido del sensor del tinaco 2
*****
*****+

int t2tiempoSenal; // tiempo de respuesta de recepcion de señal del
módulo ultrasonido
int t2litros; // variable para almacenar la converscion de volumen a
litros
float t2distanciaM; //variable que almacena el valo de la distancia
considerada como altura

//declaraciones para servomotor del tinaco uno

int servo = 7; //en este pin se activa el motorreductor del tinaco 1
int bloqueo = 0; //se almacenara el valor "1" como indicador del bloque
activo para el motor entiempos de segundos
int tope; // el valor uno indicara que el motor ya no debe girar hasta
que tenga valor cero, es para evitar que el motor gire cuando alla un
apagon

//declaraciones para servomotor del tinaco 2
*****
*****+

int t2servo = 8; //en este pin se activa el motorreductor
int t2bloqueo = 0; //se almacenara el valor "1" como indicador del bloque
activo para el motor entiempos de segundos
int t2tope; // el valor uno indicara que el motor ya no debe girar hasta
que tenga valor cero, es para evitar que el motor gire cuando alla un
apagon

//pin para el relevador que controlara la bomba de agua

```

```

int rele = 8; // en este pin se mandara la señal de activacion desde el
arduino

//Vaores para calcular el tiempo en segundos los cuales cotrolaran las
variable de bloqueo de los motorreductores

unsigned long tactual = 0;
unsigned long tpasado = 0;
unsigned long tiempo = 0;
int minutos = 0;
boolean estado = LOW;

//Variables del sensor de flujo de agua

const int sensor = 2; // Pin digital para el sensor de flujo YF-S201
int litros_Hora; // Variable que almacena el caudal (L/hora)
volatile int pulsos = 0; // Variable que almacena el número de pulsos
unsigned long tiempoAnterior = 0; // Variable para calcular el tiempo
transcurrido
unsigned long pulsos_Acumulados = 0; // Variable que almacena el número
de pulsos acumulados
//float litros; // // Variable que almacena el número de litros
acumulados

// Rutina de servicio de la interrupción (ISR)del flujometro
void flujo()
{
  pulsos++; // Incrementa en una unidad el número de pulsos
}

void setup() {
  Wire.begin(); // Inicia el puerto I2C
  lcd.init(); //Se enciende la lcd
  lcd.backlight(); //Se enciende la luz retractil de la lcd, para ajustar
el brillo de la misma se debe hacer desde la interface I2C

  //se asignan las funciones de salidad para los pines de los sensores de
ultrasonido

  //Tinaco uno
  pinMode(pin_TRIG, OUTPUT);
  pinMode(pin_ECHO, INPUT);

  //Tinaco dos

  pinMode(t2pin_TRIG, OUTPUT);
  pinMode(t2pin_ECHO, INPUT);

  //se asigna tipo de salida de los pines para los servomotores

  pinMode (servo, OUTPUT);
  pinMode (t2servo, OUTPUT);

  pinMode(rele, OUTPUT); // asigna tipo de salida para el relevador de la
bomba

```

```

{
  //funcion del flujometro

  pinMode(sensor, INPUT_PULLUP); // Pin digital como entrada con
conexión PULL-UP interna

  interrupts(); // Habilito las interrupciones
  // Interrupción INT0, llama a la ISR llamada "flujo" en cada flanco
de subida en el pin digital 2
  attachInterrupt(digitalPinToInterrupt(sensor), flujo, RISING);
  tiempoAnterior = millis(); // Guardo el tiempo que tarda el
ejecutarse el setup
}
}

void loop() {
  //se inicia conteode tiempo en segundos

  tactual = millis() / 1023;
  tiempo = tactual - tpasado;
  if (tiempo >= 59) {
    minutos = (minutos + 1);
    estado = !estado;
    tpasado = tactual;
    digitalWrite (13, estado);
  }
  {
    // se valoriza el cálculo de litros en el tinaco uno

    digitalWrite(pin_TRIG, LOW);
    delayMicroseconds(5); //Estabiliza el sensor

    digitalWrite(pin_TRIG, HIGH); //Envia el pulso para iniciar el sensor
    delayMicroseconds(10);

    tiempoSenal = pulseIn(pin_ECHO, HIGH); //Tiempo que dura el pin ECHO
en HIGH

    distanciaM = ((0.034 * tiempoSenal) / 2) / 100; //Convierte el tiempo
en distancia (m)
    litros = 1100 - (distanciaM * .5 * .5 * 3.1416 * 1000);

    lcd.setCursor(0, 0);
    lcd.print("T1 Lts: ");
    lcd.print(litros); //Muestra en la pantalla la distancia en
metros
    lcd.print(" ");
    lcd.setCursor(17, 0);
    lcd.print( EEPROM.get(0, tope) );
  }
  {
    //se valoriza el cálculo del tinaco 2
*****
*****+

    digitalWrite(t2pin_TRIG, LOW);

```

```

    delayMicroseconds(5); //Estabiliza el sensor

    digitalWrite(t2pin_TRIG, HIGH); //Envia el pulso para iniciar el
sensor
    delayMicroseconds(10);

    t2tiempoSenal = pulseIn(t2pin_ECHO, HIGH); //Tiempo que dura el pin
ECHO en HIGH

    t2distanciaM = ((0.034 * t2tiempoSenal) / 2) / 100; //Convierte el
tiempo en distancia (m)
    t2litros = 1100 - (t2distanciaM * .5 * .5 * 3.1416 * 1000);

    lcd.setCursor(0, 1);
    lcd.print("T2 Lts: ");
    lcd.print(t2litros);          //Muestra en la pantalla la distancia en
metros
    lcd.print(" ");
    lcd.setCursor(17, 1);
    lcd.print( EEPROM.get(1, t2tope) );
}
{
    // sentencia de apertura para llenado de tinaco 1
*****
*****+

    if (litros > 1000 && EEPROM.get(0, tope) == 0) { //EEPROM.get lee el
dato almacenado desde el bloque 0 de la memoria, el valor almacenado es
la que está en la variable tope
        //digitalWrite(rele, HIGH); //se enciende el rele para que la bomba
se active
        lcd.setCursor(13, 0);
        lcd.print("Cer");
    }

    else if (litros < 890 && EEPROM.get(0, tope) == 0) { //EEPROM.get lee
el dato almacenado desde el bloque 0 de la memoria, el valor almacenado
es la que está en la variable tope

        //digitalWrite(rele, LOW); //se enciende el rele para que la bomba
se active

        lcd.setCursor(13, 0);
        lcd.print("Abi");

        // {==[=====> (180°) <=====]==}
        for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
            digitalWrite (servo, HIGH);
            delayMicroseconds(500);          // llevamos a 180°
            digitalWrite (servo, LOW);
            delay(90);          //tiempo de espera en milisegundos para que gire
el motor
            bloqueo = 5;          // el valor de bloque es lo segundos que pasa en
el cronometro
            tope = 1;

```

```

        EEPROM.put(0, tope); // escribe en direccion
cero el valor de la variable tope
    }
}
else if (bloqueo == 5) { // se iniciara el apagado del motor

    // {[=====> (90°) <=====]}
    for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
        digitalWrite (servo, HIGH);
        delayMicroseconds(1500); // llevamos a 90°
        digitalWrite (servo, LOW);
        delay(19);

    }
    {
        if (litros > 890 && litros < 1000) {
            bloqueo = 0;
        }
    }
}

// sentencia de cierre por llenado de tinaco uno
*****
*****+

    else if (litros < 890 && EEPROM.get(0, tope) == 1) { //EEPROM.get lee
el dato almacenado desde el bloque 0 de la memoria, el valor almacenado
es la que está en la variable tope
        //digitalWrite(rele, LOW); //se enciende el rele para que la bomba
se active
        lcd.setCursor(13, 0);
        lcd.print("Abi");
    }

    else if (litros > 1000 && EEPROM.get(0, tope) == 257) { //EEPROM.get
lee el dato almacenado desde el bloque 0 de la memoria, el valor
almacenado es la que está en la variable tope
        //digitalWrite(rele, HIGH); // se da la instruccion de apagar la
bomba

        lcd.setCursor(13, 0);
        lcd.print("Cer");

        // {[=====> (180°) <=====]}
        for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
            digitalWrite (servo, HIGH);
            delayMicroseconds(2300); // llevamos a 180°
            digitalWrite (servo, LOW);
            delay(90);
            bloqueo = 5;
            tope = 0;
            EEPROM.put(0, tope); // escribe en direccion
cero el valor de la variable tope
        }
    }
}

```

```

else if (bloqueo == 5) { // se iniciara el apagado del motor

    // {[=====> (90°) <=====]}
    for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
        digitalWrite (servo, HIGH);
        delayMicroseconds(1500); // llevamos a 90°
        digitalWrite (servo, LOW);
        delay(19);

    }
    {
        if (litros > 890 && litros < 1000) {
            bloqueo = 0;

        }
    }
}

{
    // sentencia de apertura para llenado de tinaco 2
    *****
    *****+

    if (t2litros > 1000 && EEPROM.get(1, t2tope) == 0) { //EEPROM.get lee
el dato almacenado desde el bloque 0 de la memoria, el valor almacenado
es la que está en la variable tope
        //digitalWrite(rele, HIGH); //se enciende el rele para que la bomba
se active
        lcd.setCursor(13, 1);
        lcd.print("Cer");
    }

    else if (t2litros < 890 && EEPROM.get(1, t2tope) == 0) { //EEPROM.get
lee el dato almacenado desde el bloque 0 de la memoria, el valor
almacenado es la que está en la variable tope

        //digitalWrite(rele, LOW); //se enciende el rele para que la bomba
se active

        lcd.setCursor(13, 1);
        lcd.print("Abi");

        // {[=====> (180°) <=====]}
        for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
            digitalWrite (t2servo, HIGH);
            delayMicroseconds(500); // llevamos a 180°
            digitalWrite (t2servo, LOW);
            delay(90); //tiempo de espera en milisegundos para que gire
el motor
            t2bloqueo = 5; // el valor de bloque es lo segundos que pasa en
el cronometro
            t2tope = 1;
            EEPROM.put(1, t2tope); // escribe en
direccion cero el valor de la variable tope

```

```

    }
  }
  else if (t2bloqueo == 5) { // se iniciara el apagado del motor

    // {[=====> (90°) <=====]}
    for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
      digitalWrite (t2servo, HIGH);
      delayMicroseconds(1500); // llevamos a 90°
      digitalWrite (t2servo, LOW);
      delay(19);

    }
    {
      if (t2litros > 890 && t2litros < 1000) {
        t2bloqueo = 0;
      }
    }
  }

  // sentencia de cierre por llenado de tinaco 2
  *****
  *****+

  else if (t2litros < 890 && EEPROM.get(1, t2tope) == 1) { //EEPROM.get
lee el dato almacenado desde el bloque 0 de la memoria, el valor
almacenado es la que esta en la variable tope
  // digitalWrite(rele, LOW); //se enciende el rele para que la bomba
se active
    lcd.setCursor(13, 1);
    lcd.print("Abi");
  }

  else if (t2litros > 1000 && EEPROM.get(1, t2tope) == 1) {
//EEPROM.get lee el dato almacenado desde el bloque 0 de la memoria, el
valor almacenado es la que esta en la variable tope
  //digitalWrite(rele, HIGH); // se da la instruccion de apagar la
bomba

    lcd.setCursor(13, 1);
    lcd.print("Cer");

    // {[=====> (180°) <=====]}
    for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
      digitalWrite (t2servo, HIGH);
      delayMicroseconds(2300); // llevamos a 180°
      digitalWrite (t2servo, LOW);
      delay(90);
      t2bloqueo = 5;
      t2tope = 0;
      EEPROM.put(1, t2tope); // escribe en
direccion cero el valor de la variable tope
    }
  }
  else if (t2bloqueo == 5) { // se iniciara el apagado del motor

```

```
    // {[=====> (90°) <====]=}
    for (int Hz = 0; Hz < 50 ; Hz++) { // repetimos la instruccion 50
veces
        digitalWrite (t2servo, HIGH);
        delayMicroseconds(1500);           // llevamos a 90°
        digitalWrite (t2servo, LOW);
        delay(19);

    }
    {
        if (t2litros > 890 && t2litros < 1000) {
            t2bloqueo = 0;

        }
    }
}
}
}
```