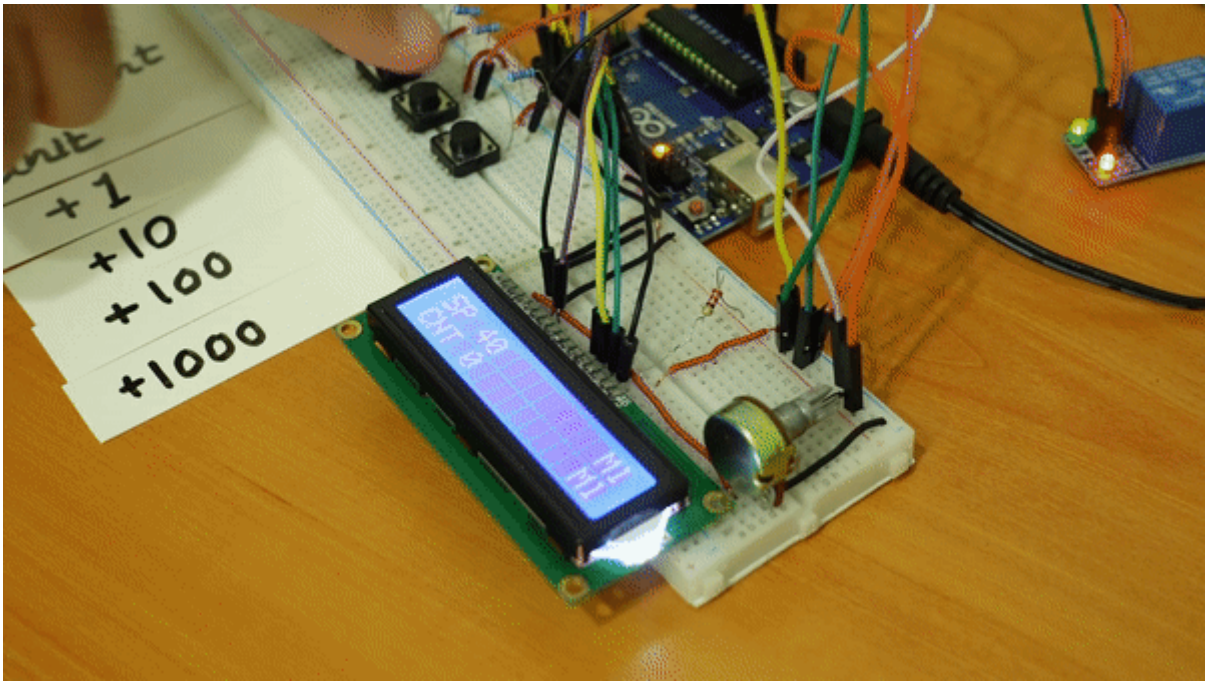
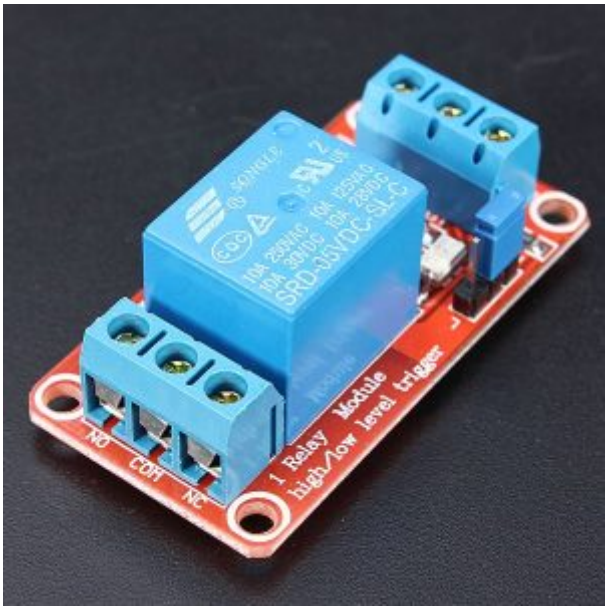
مقياس تدفق الماء و التحكم بالكمية باستخدام الاردوينو

من وسائل ترشيد استخدام المياه توفير أنظمة ذكية تقيس وتحدد كمية المياه، من خلال هذا الدرس يمكنك عمل نظام يحدد الكمية المطلوبة من الماء اما بوحدة المليلتر أو اللتر.
النظام يحتوي على الأردوينو كوحدة تحكم وحساس التدفق لقياس تدفق الماء وحساب الكمية، ومضخة يتم اغلاقها وتشغيلها حسب اعدادات النظام، وكذلك يتكون النظام من شاشة كرستالية كواجهة للمستخدم عند تحديد الخيارات، وسيتم التحكم بالنظام باستخدام ضغاط التحكم لتغيير الإعدادات.



المواد و الأدوات*

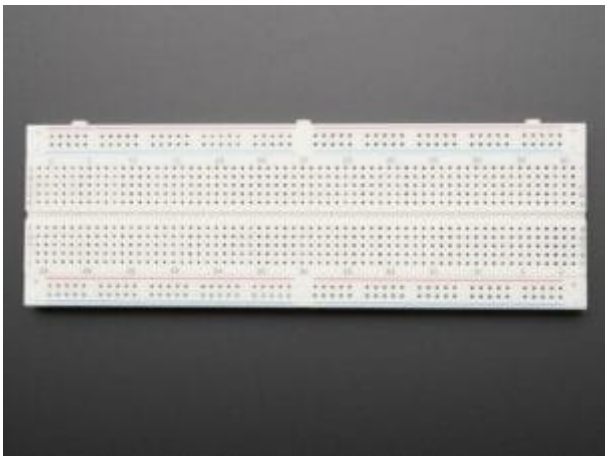
X1 أردوينو أونو



X1 مرحل

X1 سلك الأردوينو

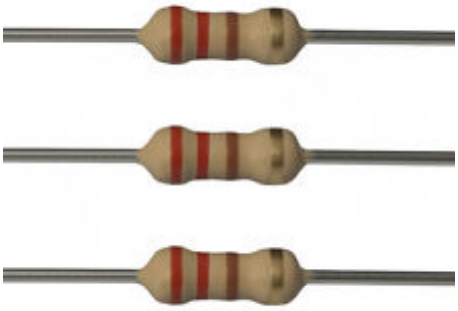
X8 ضغاط التحكم



X2 لوحة تجارب

X1 حساس التدفق

X8 مقاومة 10 كيلو أوم



1X مقاومة 220 أوم

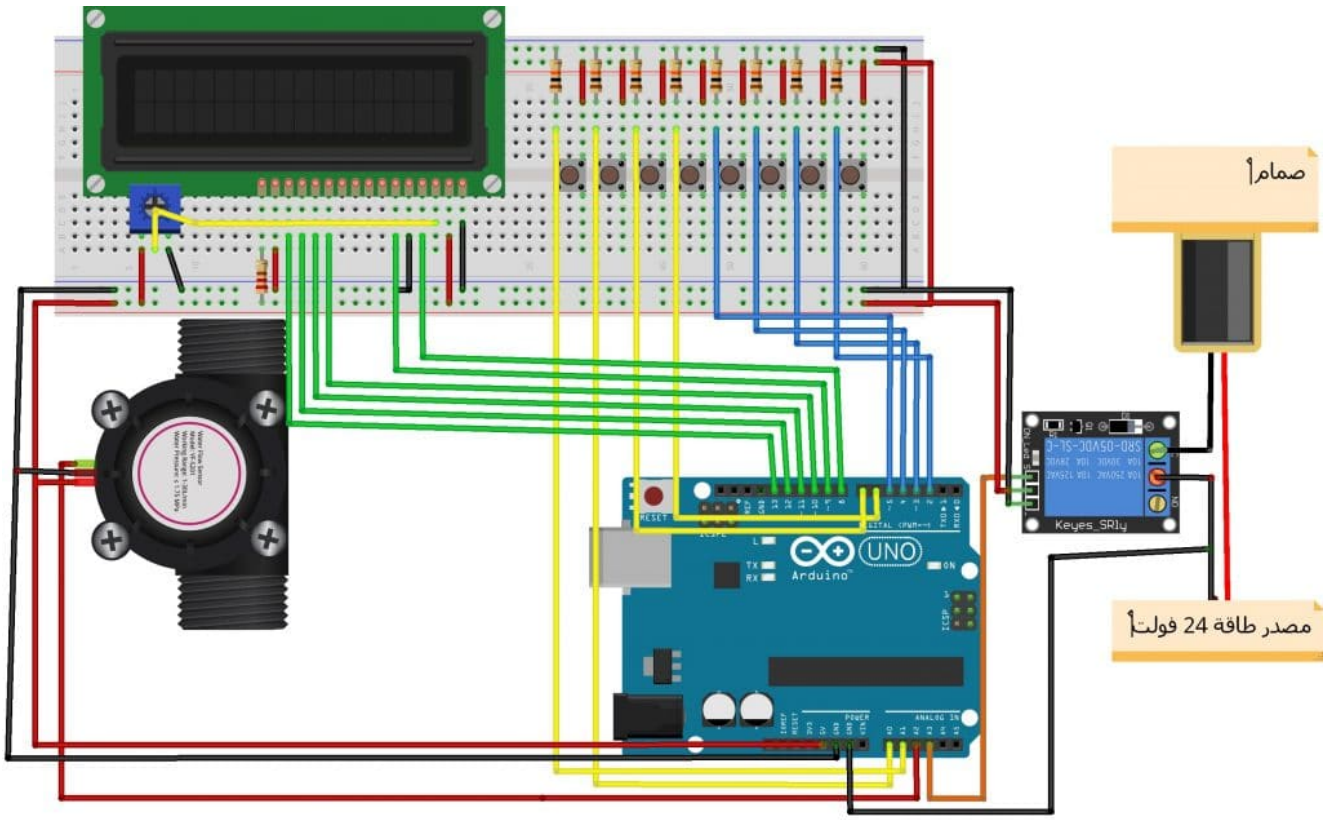
X1 مقاومة متغيرة

X1 ملف صمام

X1 وصلة تيار ثابت

X1 مصدر طاقة 24 فولت

توصيل الدائرة



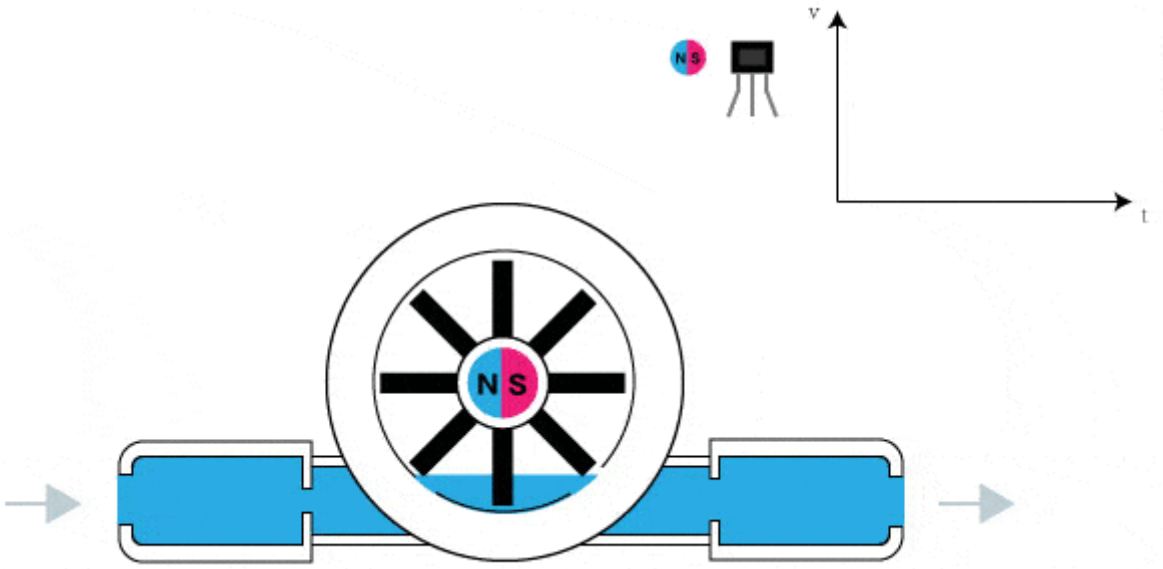
حساس التدفق

نستخدم حساس التدفق لقياس معدل تدفق المياه أو اي سائل آخر. معدل تدفق الماء هو حجم السائل الذي يمر في كل وحدة زمنية. من أشهر تطبيقات مستشعر التدفق للتحكم التلقائي في سخانات المياه، وآلات صنع القهوة، وآلات بيع المياه، وما إلى ذلك.

يعتمد مستشعر التدفق مبدأ تأثير هول (Hall).

حيث يتكون المستشعر من مستشعر تأثير القاعدة (Hall Effect) وعجلة التوربين والمغناطيس. يتدفق الماء من خلال المدخل ويخرج من خلال المخرج. دفع تيار الماء العجلة إلى الدوران، وتدار معها المغناطيس الموجود على العجلة. يؤدي دوران المجال المغناطيسي إلى تشغيل مستشعر القاعة، والذي ينتج موجات مربعة عالية ومنخفضة المستوى (نبض).

لكل جولة من العجلة، يكون حجم المياه المتدفقة مقدارًا معينًا، وكذلك عدد الموجات المربعة الناتجة. لذلك، يمكننا حساب تدفق المياه عن طريق حساب عدد الموجات المربعة (النبض).



ويحتوي المستشعر على 3 أسلاك باللون الأحمر والأصفر والأسود. يستخدم السلك الأحمر للجهد الذي يتراوح من 5 فولت إلى 18 فولت والسلك الأسود متصل بـ GND. يستخدم السلك الأصفر للإخراج (نبضات)، والتي يمكن قراءتها بواسطة المتحكم ، كذلك يظهر على المستشعر سهم يشير إلى مسار الماء

فكرة المشروع

النظام في هذا المشروع يتكون من 8 مفاتيح تحكم و شاشة عرض كرستالية، بالإضافة لحساس التدفق، الذي يتم من خلاله قياس كمية الماء، وكذلك يوجد صمام الملف الذي يسمح بمرور الماء إلى الحساس، و يتم غلق الصمام عند وصول الماء للمستوى المطلوب.
وظائف المفاتيح بالنظام كالتالي :

أولا: مفتاح تشغيل النظام، بعد تحديد الكمية الضغط عليه سيؤدي إلى تشغيل صمام الملف أو إيقافه. عندما يكون هذا المفتاح في حالة تشغيل ستوقف النظام ولن تعمل أي مفاتيح أخرى.

ثانيًا: مفتاح تحديد الوحدة يمكن قياس الكمية بوحدة مليلتر أو لتر، إذا كان في حالة 0يعني أنه على وحدة مليلتر. وحالة 1 يعني لتر

ثالثا: متاح إعادة ضبط الكمية المعينة إلى 0، حيث تُظهر الشاشة SP والتي تعني الكمية التي يعمل النظام للوصول لها والتي تم تحديدها من قبل المستخدم.

رابعًا: زر إعادة ضبط عداد الكمية المعبئة إلى 0، لبدء تعبئة جديدة، حيث تظهر الشاشة CNT والتي تشير لكمية الماء التي تم تعبئتها

خامسًا: مفاتيح تعين كمية الماء، وهي أربع مفاتيح +1,+10,+100,+1000.

تحديد عدد نبضات لتر واحد من الماء

مواصفات المنتج أن 450 نبضة للتر واحد من الماء لكن للوصول لقراءة دقيقة سنقوم بعملية الوزن ،عن طريق أخذ قراءة عدد النبضات عند تمرير لتر واحد من الماء

الكود البرمجي

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //Pines arduino to lcd

//-------Pins-----//
int Relay = 1; //Solenoid valve open/close
int start_stop = 2; //Start/Stop button
const int sensor_pulse = 16; //Sensor pulse
int rst_cnt = 3; // Reset counter button
//---------Storage debounce function-----//
boolean currentstart_stop = LOW;
boolean laststart_stop =LOW;
boolean lastsensor_pulse = LOW;
boolean currentsensor_pulse = LOW;
boolean lastrst_cnt = LOW;
boolean currentrst_cnt = LOW;
boolean RelayState = LOW;

int counter = 0;

void setup() {
pinMode(Relay, OUTPUT);
lcd.begin(16, 2);
lcd.setCursor(5, 0);
lcd.print("COUNTER");
lcd.setCursor(0, 1);
lcd.print("PULSES");

}
//----Debouncing function----//
boolean debounce(boolean last, int pin)
{
boolean current = digitalRead(pin);
if (last != current)
{
delay(5);
current = digitalRead(pin);
}
return current;
}
```

```
void loop() {
currentstart_stop = debounce(laststart_stop, start_stop); //Debounce for Start/Stop
Button
currentsensor_pulse = debounce(lastsensor_pulse, sensor_pulse); //Debounce for
Sensor pulse Button
currentrst_cnt = debounce(lastrst_cnt, rst_cnt); //Debounce for reset counter Button

//-----Start/Stop toggle function----//
if (currentstart_stop == HIGH && laststart_stop == LOW){

if (RelayState == HIGH){ //Toggle the state of the Relay
digitalWrite(Relay, LOW);
RelayState = LOW;


}
else{
digitalWrite(Relay, HIGH);
RelayState = HIGH;
}
}

laststart_stop = currentstart_stop;

if (lastsensor_pulse== LOW && currentsensor_pulse == HIGH){
counter=counter+ 1;
}
lastsensor_pulse = currentsensor_pulse;

lcd.setCursor(7, 1);
lcd.print(counter);

if(RelayState == LOW){ //Reset counter while sistem is not running
if (currentrst_cnt == HIGH && lastrst_cnt == LOW){//Reset Counter
lcd.setCursor(6, 1); // Clear CNT area
lcd.print(" ");
counter= 0;

}
lastrst_cnt = currentrst_cnt;
}

}// end void loop
```

شرح الكود البرمجي

استدعاء مكتبة الشاشة الكرستالية، و تعريف المنافذ للشاشة

```
 #include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //Pines arduino to lcd
```

تعريف منافذ الحساس تم توصيله مع منفذ A2= 16، و تسمية المتغير sensor_pulse، كذلك تعريف منفذ مفتاح التحكم بتشغيل

النظام و هو المنفذ رقم 3 و منفذ مفتاح التحكم بتصفير العداد و هو المنفذ رقم 2

```
//-------Pins-----//
int Relay = 17; //Solenoid valve open/close
int start_stop = 2; //Start/Stop button
const int sensor_pulse = 16; //Sensor pulse
int rst_cnt = 3; // Reset counter button
```

تعين الحالة الابتدائية للمتغرات

```
boolean currentstart_stop = LOW;
boolean laststart_stop =LOW;
boolean lastsensor_pulse = LOW;
boolean currentsensor_pulse = LOW;
boolean lastrst_cnt = LOW;
boolean currentrst_cnt = LOW;
boolean RelayState = LOW;

int counter = 0;
```

في دالة void setup نقوم بتهيئة الشاشة وتعريف منافذ الادخال و الاخراج

```
void setup() {
pinMode(Relay, OUTPUT);
lcd.begin(16, 2);
lcd.setCursor(5, 0);
lcd.print("COUNTER");
lcd.setCursor(0, 1);
lcd.print("PULSES");

}
//----Debouncing function----//
boolean debounce(boolean last, int pin)
{
boolean current = digitalRead(pin);
if (last != current)
{
delay(5);
current = digitalRead(pin);
}
return current;
}
```

في دالة void loop يتم حساب عدد النبضات و عرضها على الشاشة الكرستالية

```
void loop() {
currentstart_stop = debounce(laststart_stop, start_stop); //Debounce for Start/Stop
Button
currentsensor_pulse = debounce(lastsensor_pulse, sensor_pulse); //Debounce for
Sensor pulse Button
currentrst_cnt = debounce(lastrst_cnt, rst_cnt); //Debounce for reset counter Button
```

```
//-----Start/Stop toggle function----//
if (currentstart_stop == HIGH && laststart_stop == LOW){

if (RelayState == HIGH){ //Toggle the state of the Relay
digitalWrite(Relay, LOW);
RelayState = LOW;


}
else{
digitalWrite(Relay, HIGH);
RelayState = HIGH;
}
}

laststart_stop = currentstart_stop;
if (lastsensor_pulse== LOW && currentsensor_pulse == HIGH){
counter=counter+ 1;
}
lastsensor_pulse = currentsensor_pulse;
lcd.setCursor(7, 1);
lcd.print(counter);
if(RelayState == LOW){ //Reset counter while sistem is not running
if (currentrst_cnt == HIGH && lastrst_cnt == LOW){//Reset Counter
lcd.setCursor(6, 1); // Clear CNT area
lcd.print(" ");
counter= 0;
}
lastrst_cnt = currentrst_cnt;
}
}// end void loop
```

قم بحفض القيمة لكل لتر من الماء ليتم تعديلها في الشفرة البرمجية للمشروع

الكود البرمجي

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //Pines arduino to lcd

//-------Pins-----//
int Relay = 17; //Solenoid valve open/close
int start_stop = 2; //Start/Stop button
int rst_sp = 3; // Reset Set Point Button
int rst_cnt = 4; // Reset counter button
int unit = 5; // Change Unit Button
const int sensor_pulse =16; // Sensor Pulse In
//----Analog as Input-----//
int add_one =6; // +1 Button
int add_ten = 7; // +10 Button
int add_cien = 14; // +100 Button
```

```
int add_mil = 15; // +1000 Buton

//-----Variables for debouncing-----//
boolean currentstart_stop = LOW;
boolean laststart_stop =LOW;
boolean lastsensor_pulse = LOW;
boolean currentsensor_pulse = LOW;
boolean lastunit = LOW;
boolean currentunit = LOW;
boolean lastrst_sp = LOW;
boolean currentrst_sp = LOW;
boolean lastrst_cnt = LOW;
boolean currentrst_cnt = LOW;
boolean lastadd_one = LOW;
boolean currentadd_one = LOW;
boolean lastadd_ten = LOW;
boolean currentadd_ten = LOW;
boolean lastadd_cien = LOW;
boolean currentadd_cien = LOW;
boolean lastadd_mil = LOW;
boolean currentadd_mil = LOW;

//-----Storage state for toggle function---//
boolean unitState = LOW; //storage for the current state of the unit
boolean RelayState = LOW; //storage for the current state of the Relay (off/on)

//-------You have to put your pulses x liters here-----//
float cal_1=2.5; //Calibrate ml x pulse (cal_1 = 1000/400)
int cal_2= 400; //Calibrate pulses x liters
//-----------------------------------------------------//

float counter_1 = 0.0;
int counter_2= 0;
int TotalCount_1= 0;
int TotalCount_2= 0;
int set_point_1= 0;
int set_point_2= 0;

void setup(){
lcd.begin(16, 2);
pinMode(Relay, OUTPUT);
pinMode(add_one, INPUT);
pinMode(add_ten, INPUT);
pinMode(add_cien, INPUT);
pinMode(add_mil, INPUT);

lcd.setCursor(0,0); //Show "SP" on the LCD
lcd.print("SP");
lcd.setCursor(0,1); //Show "CNT" on the LCD
lcd.print("CNT");

}
//----Debouncing function for all buttons----//
```

```
boolean debounce(boolean last, int pin)
{
boolean current = digitalRead(pin);
if (last != current)
{
delay(5);
current = digitalRead(pin);
}
return current;
}

void loop(){
//-----Debounce Buttons-----//
currentstart_stop = debounce(laststart_stop, start_stop); //Debounce for Start/Stop
Button
currentsensor_pulse = debounce(lastsensor_pulse, sensor_pulse); //Debounce for
Sensor
currentunit = debounce(lastunit, unit); //Debounce for unit Button
currentrst_sp = debounce(lastrst_sp, rst_sp); //Debounce for reset set point Button
currentrst_cnt = debounce(lastrst_cnt, rst_cnt); //Debounce for reset counter Button
currentadd_one = debounce(lastadd_one, add_one); //Debounce for +1 Button
currentadd_ten = debounce(lastadd_ten, add_ten); //Debounce for +10 Button
currentadd_cien = debounce(lastadd_cien, add_cien); //Debounce for +100 Button
currentadd_mil = debounce(lastadd_mil, add_mil); //Debounce for +1000 Button


//-----Start/Stop toggle function----//
if (currentstart_stop == HIGH && laststart_stop == LOW){

if (RelayState == HIGH){ //Toggle the state of the Relay
digitalWrite(Relay, LOW);
RelayState = LOW;
}
else{
digitalWrite(Relay, HIGH);
RelayState = HIGH;
}
}

laststart_stop = currentstart_stop;

//-------Unit toggle function----//
if(RelayState == LOW){ //You only can change unit while system is not running!

//------ Lt/ml unit toggle function----//
if (currentunit == HIGH && lastunit == LOW){
lcd.setCursor(4, 1); //Clear lcd(CNT area) between unit change,keeping last count
lcd.print(" ");
lcd.setCursor(3,0); //Clear lcd (SP area) between unit change, keeping last SP
lcd.print(" ");

if (unitState == HIGH){ //Toggle the state of the unit (L/ml)
digitalWrite(unit, LOW);
```

```
unitState = LOW;
}
else{
digitalWrite(unit, HIGH);
unitState = HIGH;
}
}
lastunit = currentunit;
}
//------Print unit state-----//
if(unitState==HIGH){ //Unit state HIGH = L
lcd.setCursor(14,0);
lcd.print("Lt");
lcd.setCursor(14, 1);
lcd.print("Lt");

}
else { //Unit state LOW = ml
lcd.setCursor(14,0);
lcd.print("Ml");
lcd.setCursor(14,1);
lcd.print("Ml");
}//End Print unit state

//--------------------------//
//------------Ml Counter-----//
//--------------------------//
if(unitState==LOW){ // LOW= Ml state

//----------------------//
//-----Settings----------//
//----------------------//

if(RelayState == LOW){ // You only can change settings while system is not running!

//-----Adders Buttons (set point_1)---//

if (currentadd_ten == HIGH && lastadd_ten == LOW){ // Add +10
set_point_1 = set_point_1 +10;
}
lastadd_ten = currentadd_ten;

if (currentadd_cien == HIGH && lastadd_cien == LOW){ // Add +100
set_point_1 = set_point_1 +100;
}
lastadd_cien = currentadd_cien;

if (currentadd_mil == HIGH && lastadd_mil == LOW){ // Add +1000
set_point_1 = set_point_1 +1000;
}
lastadd_mil = currentadd_mil;

//-------Reset Buttons----//
```

```
if (currentrst_sp == HIGH && lastrst_sp == LOW){ //Reset Set Point
lcd.setCursor(3, 0); // Clear SP area
lcd.print(" ");
set_point_1 = 0;
}
lastrst_sp = currentrst_sp;
if (currentrst_cnt == HIGH && lastrst_cnt == LOW){//Reset Counter
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_1= 0;
TotalCount_1= 0;
}
lastrst_cnt = currentrst_cnt;
}//-----End Settings-----//

//----Start Counter------//
if(RelayState == HIGH){ // Only counts while relay is HIGH
if (lastsensor_pulse== LOW && currentsensor_pulse == HIGH){
counter_1 = counter_1 + cal_1;
}
}
lastsensor_pulse = currentsensor_pulse;

//-------Counter function-----//
if(counter_1 >= 10){
TotalCount_1 = TotalCount_1 + 10;
counter_1=0; //Counter reset
}

lcd.setCursor(3, 0); //Show set point
lcd.print(set_point_1);
lcd.setCursor(4, 1); // Show counter
lcd.print(TotalCount_1);

//--Stop Counter.You can´t start if set point is lower or equal to counter--//
if(RelayState==HIGH){
if(set_point_1 <= TotalCount_1){
RelayState = LOW;
digitalWrite(Relay, LOW);
//***********************************************Autoreset
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_1= 0;
TotalCount_1= 0;
}
}
}//End unit state LOW (ml)

//--------------------------//
//------------Lt Counter-----//
//--------------------------//

if(unitState== HIGH){ //HIGH = Lt state
```

```
//-----------------------//
//-----Settings----------//
//-----------------------//

if(RelayState == LOW){ // You only can change settings while system is not running!

//-----Adders Buttons (set point_2)---//
if (currentadd_one == HIGH && lastadd_one == LOW){ // Add +1
set_point_2 = set_point_2 +1;
}
lastadd_one = currentadd_one;

if (currentadd_ten == HIGH && lastadd_ten == LOW){ // Add +10
set_point_2 = set_point_2 +10;
}
lastadd_ten = currentadd_ten;

if (currentadd_cien == HIGH && lastadd_cien == LOW){ // Add +100
set_point_2 = set_point_2 +100;
}
lastadd_cien = currentadd_cien;

if (currentadd_mil == HIGH && lastadd_mil == LOW){ // Add +1000
set_point_2 = set_point_2 +1000;
}
lastadd_mil = currentadd_mil;

//-------Reset Buttons----//
if (currentrst_sp == HIGH && lastrst_sp == LOW){ //Reset Set Point
lcd.setCursor(3, 0); // Clear SP area
lcd.print(" ");
set_point_2 = 0;
}
lastrst_sp = currentrst_sp;
if (currentrst_cnt == HIGH && lastrst_cnt == LOW){//Reset Counter
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_2= 0;
TotalCount_2= 0;
}
lastrst_cnt = currentrst_cnt;
}//-----End Settings-----//

//----Start Counter------//
if(RelayState == HIGH){ // Only counts while relay is HIGH
if (lastsensor_pulse== LOW && currentsensor_pulse == HIGH){
counter_2 = counter_2 + 1;
}
}
lastsensor_pulse = currentsensor_pulse;

//-------Counter function-----//
if(counter_2 == cal_2){
```

```
TotalCount_2 = TotalCount_2 + 1;
counter_2 = 0; //Counter reset
}

lcd.setCursor(3, 0); //Show set point
lcd.print(set_point_2);
lcd.setCursor(4, 1); // Show counter
lcd.print(TotalCount_2);

//--Stop Counter.You can´t start if set point is lower or equal to counter--//
if(RelayState==HIGH){
if(set_point_2 <= TotalCount_2){
RelayState = LOW;
digitalWrite(Relay, LOW);
//*****************************Autoreset
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_2= 0;
TotalCount_2= 0;
}
}


}//End unit state HIGH (L)




}//End Void Loop
```

شرح الكود البرمجي

استدعاء مكتبة الشاشة الكرستالية و تعريف منافذها

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //Pines arduino to lcd
```

تعريف منافذ مفاتيح النظام

```
//-------Pins-----//
int Relay = 17; //Solenoid valve open/close
int start_stop = 2; //Start/Stop button
int rst_sp = 3; // Reset Set Point Button
int rst_cnt = 4; // Reset counter button
int unit = 5; // Change Unit Button
const int sensor_pulse =16; // Sensor Pulse In
//----Analog as Input-----//
int add_one =6; // +1 Button
int add_ten = 7; // +10 Button
int add_cien = 14; // +100 Button
```

```
int add_mil = 15; // +1000 Buton
```

تعريف متغيرات

```
//-----Variables for debouncing-----//
boolean currentstart_stop = LOW;
boolean laststart_stop =LOW;
boolean lastsensor_pulse = LOW;
boolean currentsensor_pulse = LOW;
boolean lastunit = LOW;
boolean currentunit = LOW;
boolean lastrst_sp = LOW;
boolean currentrst_sp = LOW;
boolean lastrst_cnt = LOW;
boolean currentrst_cnt = LOW;
boolean lastadd_one = LOW;
boolean currentadd_one = LOW;
boolean lastadd_ten = LOW;
boolean currentadd_ten = LOW;
boolean lastadd_cien = LOW;
boolean currentadd_cien = LOW;
boolean lastadd_mil = LOW;
boolean currentadd_mil = LOW;

//-----Storage state for toggle function---//
boolean unitState = LOW; //storage for the current state of the unit
boolean RelayState = LOW; //storage for the current state of the Relay (off/on)
```

هنا يتم تعديل الأمر البرمجي بناء على عدد النبضات التي وجدتها في كود الموازنة، حيث تقوم بقسمة عدد النبضات على 1000 لتعين متغير cal_1 و متغسر cal_2 يساوي عدد النبضات

```
//-------You have to put your pulses x liters here-----//
float cal_1=2.5; //Calibrate ml x pulse (cal_1 = 1000/400)
int cal_2= 400; //Calibrate pulses x liters
//-------------------------------------------------------//

float counter_1 = 0.0;
int counter_2= 0;
int TotalCount_1= 0;
int TotalCount_2= 0;
int set_point_1= 0;
int set_point_2= 0;
```

في دالة void setup يتم تهيئة الشاشة

```
void setup(){
lcd.begin(16, 2);
pinMode(Relay, OUTPUT);
pinMode(add_one, INPUT);
pinMode(add_ten, INPUT);
pinMode(add_cien, INPUT);
```

```
pinMode(add_mil, INPUT);

lcd.setCursor(0,0); //Show "SP" on the LCD
lcd.print("SP");
lcd.setCursor(0,1); //Show "CNT" on the LCD
lcd.print("CNT");

}
//----Debouncing function for all buttons----//
boolean debounce(boolean last, int pin)
{
boolean current = digitalRead(pin);
if (last != current)
{
delay(5);
current = digitalRead(pin);
}
return current;
}
```

دالة الvoid loop لبدء النظام

```
void loop(){
//-----Debounce Buttons-----//
currentstart_stop = debounce(laststart_stop, start_stop); //Debounce for Start/Stop
Button
currentsensor_pulse = debounce(lastsensor_pulse, sensor_pulse); //Debounce for
Sensor
currentunit = debounce(lastunit, unit); //Debounce for unit Button
currentrst_sp = debounce(lastrst_sp, rst_sp); //Debounce for reset set point Button
currentrst_cnt = debounce(lastrst_cnt, rst_cnt); //Debounce for reset counter Button
currentadd_one = debounce(lastadd_one, add_one); //Debounce for +1 Button
currentadd_ten = debounce(lastadd_ten, add_ten); //Debounce for +10 Button
currentadd_cien = debounce(lastadd_cien, add_cien); //Debounce for +100 Button
currentadd_mil = debounce(lastadd_mil, add_mil); //Debounce for +1000 Button

//-----Start/Stop toggle function----//
if (currentstart_stop == HIGH && laststart_stop == LOW){

if (RelayState == HIGH){ //Toggle the state of the Relay
digitalWrite(Relay, LOW);
RelayState = LOW;
}
else{
digitalWrite(Relay, HIGH);
RelayState = HIGH;
}
}
```

```
laststart_stop = currentstart_stop;

//-------Unit toggle function----//
if(RelayState == LOW){ //You only can change unit while system is not running!

//------ Lt/ml unit toggle function----//
if (currentunit == HIGH && lastunit == LOW){
lcd.setCursor(4, 1); //Clear lcd(CNT area) between unit change,keeping last count
lcd.print(" ");
lcd.setCursor(3,0); //Clear lcd (SP area) between unit change, keeping last SP
lcd.print(" ");

if (unitState == HIGH){ //Toggle the state of the unit (L/ml)
digitalWrite(unit, LOW);
unitState = LOW;
}
else{
digitalWrite(unit, HIGH);
unitState = HIGH;
}
}
lastunit = currentunit;
}
```

الشرط التالي لتغير الوحدة

```
//------Print unit state-----//
if(unitState==HIGH){ //Unit state HIGH = L
lcd.setCursor(14,0);
lcd.print("Lt");
lcd.setCursor(14, 1);
lcd.print("Lt");

}
else { //Unit state LOW = ml
lcd.setCursor(14,0);
lcd.print("Ml");
lcd.setCursor(14,1);
lcd.print("Ml");
}//End Print unit state

//--------------------------//
//------------Ml Counter-----//
//--------------------------//
if(unitState==LOW){ // LOW= Ml state

//----------------------//
//-----Settings----------//
//----------------------//
```

يمكن تغير الإعدادات اذا كان النظام لايعمل

```
if(RelayState == LOW){ // You only can change settings while system is not running!

//-----Adders Buttons (set point_1)---//

if (currentadd_ten == HIGH && lastadd_ten == LOW){ // Add +10
set_point_1 = set_point_1 +10;
}
lastadd_ten = currentadd_ten;

if (currentadd_cien == HIGH && lastadd_cien == LOW){ // Add +100
set_point_1 = set_point_1 +100;
}
lastadd_cien = currentadd_cien;

if (currentadd_mil == HIGH && lastadd_mil == LOW){ // Add +1000
set_point_1 = set_point_1 +1000;
}
lastadd_mil = currentadd_mil;
```

الشرط التالي لتحديد عمل مفتاح اعادة الضبط

```
//-------Reset Buttons----//
if (currentrst_sp == HIGH && lastrst_sp == LOW){ //Reset Set Point
lcd.setCursor(3, 0); // Clear SP area
lcd.print(" ");
set_point_1 = 0;
}
lastrst_sp = currentrst_sp;
if (currentrst_cnt == HIGH && lastrst_cnt == LOW){//Reset Counter
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_1= 0;
TotalCount_1= 0;
}
lastrst_cnt = currentrst_cnt;
}//-----End Settings-----//

//----Start Counter------//
if(RelayState == HIGH){ // Only counts while relay is HIGH
if (lastsensor_pulse== LOW && currentsensor_pulse == HIGH){
counter_1 = counter_1 + cal_1;
}
}
lastsensor_pulse = currentsensor_pulse;
```

دالة العداد

```
//-------Counter function-----//
if(counter_1 >= 10){
TotalCount_1 = TotalCount_1 + 10;
counter_1=0; //Counter reset
}
```

```
lcd.setCursor(3, 0); //Show set point
lcd.print(set_point_1);
lcd.setCursor(4, 1); // Show counter
lcd.print(TotalCount_1);

//--Stop Counter.You can´t start if set point is lower or equal to counter--//
if(RelayState==HIGH){
if(set_point_1 <= TotalCount_1){
RelayState = LOW;
digitalWrite(Relay, LOW);
//*********************************************Autoreset
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_1= 0;
TotalCount_1= 0;
}
}
}//End unit state LOW (ml)

//---------------------------//
//------------Lt Counter-----//
//---------------------------//

if(unitState== HIGH){ //HIGH = Lt state

//-----------------------//
//-----Settings----------//
//-----------------------//

if(RelayState == LOW){ // You only can change settings while system is not running!

//-----Adders Buttons (set point_2)---//
if (currentadd_one == HIGH && lastadd_one == LOW){ // Add +1
set_point_2 = set_point_2 +1;
}
lastadd_one = currentadd_one;

if (currentadd_ten == HIGH && lastadd_ten == LOW){ // Add +10
set_point_2 = set_point_2 +10;
}
lastadd_ten = currentadd_ten;

if (currentadd_cien == HIGH && lastadd_cien == LOW){ // Add +100
set_point_2 = set_point_2 +100;
}
lastadd_cien = currentadd_cien;

if (currentadd_mil == HIGH && lastadd_mil == LOW){ // Add +1000
set_point_2 = set_point_2 +1000;
}
lastadd_mil = currentadd_mil;

//-------Reset Buttons----//
```

```
if (currentrst_sp == HIGH && lastrst_sp == LOW){ //Reset Set Point
lcd.setCursor(3, 0); // Clear SP area
lcd.print(" ");
set_point_2 = 0;
}
lastrst_sp = currentrst_sp;
if (currentrst_cnt == HIGH && lastrst_cnt == LOW){//Reset Counter
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_2= 0;
TotalCount_2= 0;
}
lastrst_cnt = currentrst_cnt;
}//-----End Settings-----//

//----Start Counter------//
if(RelayState == HIGH){ // Only counts while relay is HIGH
if (lastsensor_pulse== LOW && currentsensor_pulse == HIGH){
counter_2 = counter_2 + 1;
}
}
lastsensor_pulse = currentsensor_pulse;

//-------Counter function-----//
if(counter_2 == cal_2){
TotalCount_2 = TotalCount_2 + 1;
counter_2 = 0; //Counter reset
}

lcd.setCursor(3, 0); //Show set point
lcd.print(set_point_2);
lcd.setCursor(4, 1); // Show counter
lcd.print(TotalCount_2);

//--Stop Counter.You can´t start if set point is lower or equal to counter--//
if(RelayState==HIGH){
if(set_point_2 <= TotalCount_2){
RelayState = LOW;
digitalWrite(Relay, LOW);
//*****************************Autoreset
lcd.setCursor(4, 1); // Clear CNT area
lcd.print(" ");
counter_2= 0;
TotalCount_2= 0;
}
}

}//End unit state HIGH (L)
```

```
}//End Void Loop
```