



## حساب سرعة دراجة هوائية باستخدام اردوينو

في هذا الدرس سنتعلم صنع نظام حساب سرعة دراجة هوائية باستخدام اردوينو، و مستشعر مفتاح الاتصال المغناطيسي، وسيتم عرض بيانات السرعة على شاشة كرسطالية، والنظام يعطي بيانات عن المسافة وزمن الرحلة بالدقائق، وذلك من خلال توفير مفتاح ضغط ، عند النقر على المفتاح سيتم التبديل بين شاشة تعرض المسافة و السرعة وشاشة تعرض المسافة ومدة الرحلة.



## المواد و الأدوات



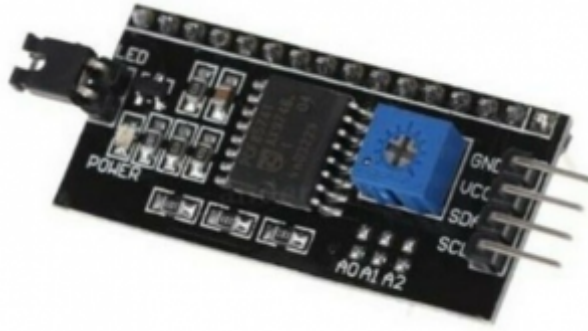
X 1 اردوينو اونو



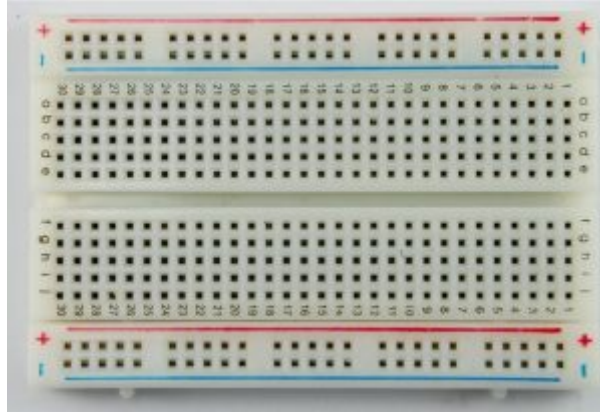
X1 سلك اردوينو



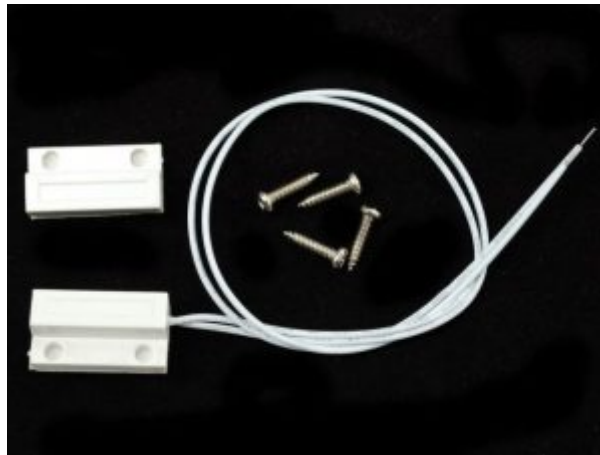
X1 شاشة كرسالية



X1 I2C / IIC Serial Interface Module



X1 لوحة تجارب



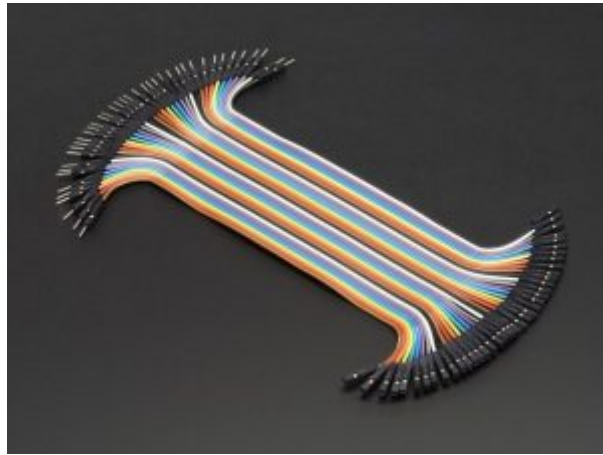
X1 مستشعر مفتاح الاتصال المغناطيسي



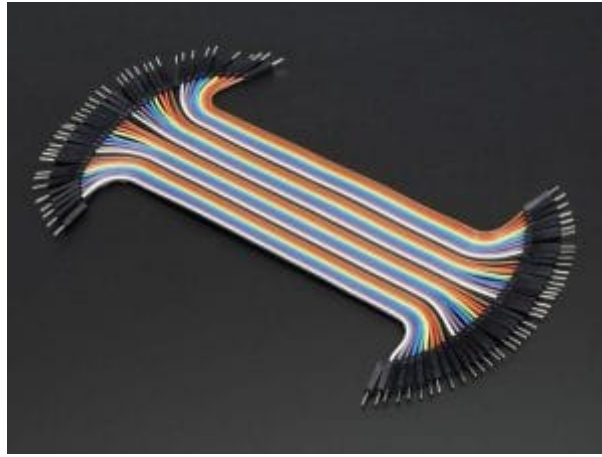
X2 مقاومة 10 كيلو أوم



X1 ضغط التحكم

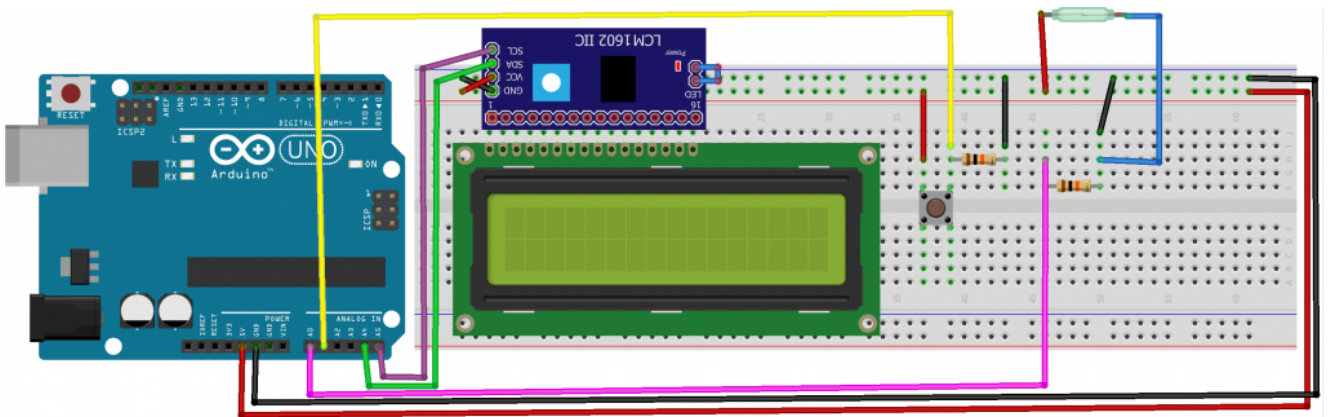


مجموعة أسلاك توصيل (M/F)



مجموعة أسلاك توصيل (M/M)

## توصيل الدائرة



## الكود البرمجي

```
//libraries
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
//definition of variables
long previous, triptime, time, impulses;
float speedometer, dist, aspeed;
int screen=1;
//If you have other circuit of wheel you need change it
float radius=2;
void setup() {
  lcd.init(); // initialize the lcd
  // Print a message to the LCD.
  lcd.setCursor(1,0);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  lcd.print("Bike speedometer");
  delay(1000);
  lcd.setCursor(5, 1);
```

```

lcd.print("V 1.0");
delay(4000);
lcd.clear();
delay(500);
lcd.setCursor(0, 0);
lcd.print("Dist:");
}
void loop() {
int aa=analogRead (A0);
Serial.println (aa);
//if wheel turns
if(analogRead(A0)>=300){
//number of turns++
impulses++;
//count turn time
time=(millis()-previous);
//count speed
speedometer=(3600000 / time) * radius/ 1000;
speedometer=(circuit / time)*3600.0;
previous=millis();
delay(100);
}
Lcd();
}
void Lcd(){
//when button is clicked
if(digitalRead (A1)==1){
lcd.clear();
screen++;
if(screen==3){
screen=1;
}
}
if(screen==1){
//displays speed
lcd.setCursor(0, 1);
lcd.print("Speed:");
lcd.setCursor(7, 1);
lcd.print(speedometer);
lcd.print("km/h");
}
if(screen== 2){
//diplays trip time
triptime=millis()/60000;
lcd.setCursor(0, 1);
lcd.print("Time:");
lcd.setCursor(7, 1);
lcd.print(triptime);
}
lcd.setCursor(0, 0);
lcd.print("Dist:");
//calculation of the distance
dist=impulses*radius/1000.00;
}
}

```

```
//dislays distance
lcd.setCursor(6,0);
lcd.print(dist);
lcd.print("km");
}
```

## شرح الكود البرمجي

سنقوم في البداية باستدعاء مكتبة (LiquidCrystal\_I2C) الخاصة بوحدة i2c والتي تحتوي على مجموعة أوامر برمجية نحتاجها في المشروع

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

نعرف عنوان وحدة i2c

```
LiquidCrystal_I2C lcd(0x27,20,4);
//definition of variables
```

نعرف المتغيرات التالية وهي (previous, triptime, time, impulses, speedometer, dist, aspeed)

```
long previous, triptime, time, impulses;
float speedometer, dist, aspeed;
```

تعريف متغير (screen) والذي يشير إلى طباعة البيانات و ستم طباعتها على 3 شاشات شاشة تحتوي على المسافة والسرعة و شاشة تعرض المسافة و متوسط السرعة وشاشة للوقت و المسافة

```
int screen=1;
```

تحديد نصف القطر للعجلة

```
float radius=2.0;
```

تهيئة الشاشة والمنافذ

```
void setup() {
  itialize the lcd
  lcd.init();
  // Print a message to the LCD.

  lcd.backlight();
  lcd.setCursor(1,0);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  lcd.print("Bike speedometer ");
  delay(1000);}
}
```

في دالة void loop يتم حساب سرعة دراجة هوائية حسب الوقت المعادلة بالدالة



```

void loop() {
int aa=analogRead (A0);
//if wheel turns
if(analogRead(A0)>=300){
//number of turns++
impulses++;
//count turn time
time=(millis()-previous);
//count speed
speedometer=(radius/ time)*3600.0;
previous=millis();
delay(100);
}

```

تحديد البيانات في كل شاشة و سيتم عرض 3 شاشات شاشة تظهر السرعة و شاشة تظهر متوسط السرعة وشاشة تظهر وقت الرحلة بالدقائق

```

Lcd();
}
void Lcd(){
//when button is clicked
if(digitalRead (A1)==1){
lcd.clear();
screen++;
if(screen==3){
screen=1;
}
}

if(screen==1){
//displays speed
lcd.setCursor(0, 1);
lcd.print("Speed:");
lcd.setCursor(7, 1);
lcd.print(speedometer);
lcd.print("km/h");
}

if(screen== 2){
//diplays trip time
triptime=millis()/60000;

lcd.setCursor(0, 1);
lcd.print("Time:");
lcd.setCursor(7, 1);
lcd.print(triptime);

}
}

```

عرض المسافة

```
lcd.setCursor(0, 0);  
lcd.print("Dist:"); //calculation of the distance  
dist=impulses*circuit/1000.00; //dislays distance  
lcd.setCursor(6,0);  
  lcd.print(dist);  
lcd.print("km"); }
```

قم بتثبيت المستشعر بحيث يكون في جزء ثابت يقابل العجلة، وعلى العجلة يتم تثبيت المغناطيس في كل مرة تتحرك العجلة و يصبح المغناطيس في مقابل الحساس سيتم حساب نبضة،

تحتاج إلى توفير صندوق باستخدام طابعة ثلاثية الأبعاد أو قاطع الليزر لحفظ كافة القطع بشكل ثابت أثناء الرحلة