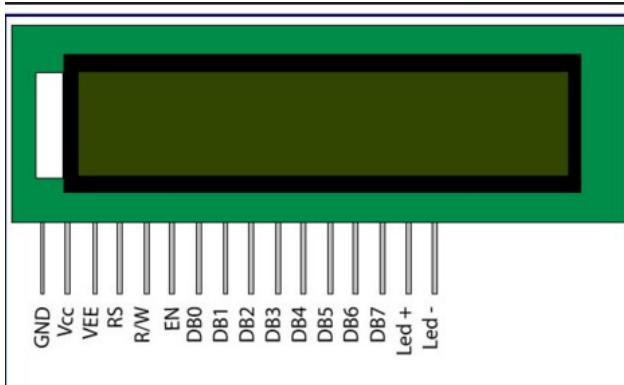


الـ LCD لها 16 pins كما بالشكل (٢. ٤٢) وبيانهم كالتالي:



شكل (٢. ٤٢)

- (١) **Vss** : يتم توصيلها بالـ **ground**.
- (٢) **Vdd** : يتم توصيلها بالـ **5V**، وهما المسؤولان عن تشغيل الشاشة.
- (٣) **Vee** : مسئولة عن سطوع الشاشة، فإذا تم توصيل الشاشة بالـ **ground** وكانت الإضاءة جيدة فيتم توصيلها بالـ **ground**، أما إذا كانت الإضاءة تحتاج إلي تعديل يتم توصيلها بمقاومة متغيرة.
- (٤) **Rs (register select)** : لمعرفة ما إذا كان الكود المرسل إليها أمر أم **data**، ويتم توصيلها بـ **digital pin** من الأردوينو.
- (٥) **R/ W** : يتم توصيلها بالـ **ground**، وتعتبر عن أننا هل سنقرأ من علي الشاشة أم هنبعث عليها؟
- (٦) **EN (Enable)** : ويتم توصيلها بـ **digital pin** من الأردوينو.
- (٧) **D0** : لا يتم توصيلها.
- (٨) **D1** : لا يتم توصيلها.
- (٩) **D2** : لا يتم توصيلها.
- (١٠) **D3** : لا يتم توصيلها.
- (١١) **D4** : **data pin** ، ويتم توصيلها بـ **digital pin** من الأردوينو.
- (١٢) **D5** : **data pin** ، ويتم توصيلها بـ **digital pin** من الأردوينو.
- (١٣) **D6** : **data pin** ، ويتم توصيلها بـ **digital pin** من الأردوينو.
- (١٤) **D7** : **data pin** ، ويتم توصيلها بـ **digital pin** من الأردوينو.
- (١٥) توصّل بـ **5V**.
- (١٦) توصّل بالـ **ground**.

معروض الأهر للتطبيقات الهندسية

برمجة الـ LCD :

-إذا أردنا أن نكتب حرف أو كلمة أو رقم علي الـ **LCD**، فلا بد من برمجتها لعمل ذلك، إذن لابد من أن هناك أوامر برمجية خاصة بها وهي كالآتي:
هناك 6 أوامر مهمة هي الأكثر استخداماً عند برمجة الـ **LCD** عن طريق الأردوينو.
ولاستخدام هذه الأوامر يتم استدعاء مكتبة الـ **LCD** لكي يقدر الأردوينو علي التعرف علي هذه الأوامر الستة، ويتم استدعاؤها عن طريق الأمر:

```
#include<LiquidCrystal.h>
```

ثم بعد ذلك نستخدم أوامر الـ **LCD** وهي مفصلة كالآتي :

- ١- **LiquidCrystal lcd(Rs,EN,D4,D5,D6,D7)** : يتم تسميته الشاشة لأننا بعد ذلك نستخدم هذا الاسم مع بقية الأوامر ، هنا تم تسميتها بـ **LCD**، وداخل القوسين نضع أرقام الـ **digital pins** التي يتم توصيلها بـ **LCD pins**.
- ٢- **Lcd.begin(cols,rows);** : يتم كتابة نوع الشاشة المستخدمة، من حيث عدد الصفوف والأعمدة.
- ٣- **Lcd.clear();** : يستخدم لمسح أي كلام مكتوب علي الشاشة.

- ٤- **Lcd.print ();** : يتم كتابة الجمل المراد طباعتها علي الشاشة، إذا أردنا طباعة كلمة او جملة نضعها بين علامتين التنصيص " "، وإذا أردنا أن نطبع حرف يتم كتابته بين ' '، وإذا أردنا طباعة رقم لا نضع أي أقواس تتم كتابة الرقم مباشرة .
- ٥- **Lcd.print (millis()/1000);** : يستخدم هذا الامر لعرض عدد الثواني علي الشاشة منمنذ بداية عمل الشاشة.
- ٦- **Lcd.setCursor (cols, rows);** : إذا أردنا أن ننتقل من مكان لآخر علي الشاشة، أو بمعني آخر لصف أو عمود آخر نضع رقم الصف والعمود المراد الانتقال إليهما ، مع العلم أن صفوف الشاشة تبدأ من رقم 0 إلي رقم 1 وكذلك الأعمدة تبدأ من رقم 0 إلي رقم 15.

و الشكل (٤٣.٢) يوضح كود لعرض كلمة **AZEX'16** علي الشاشة ...

```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("AZEX'16");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

شكل (٤٣.٢)

وإلي هنا نكون قد انتهينا من جزء الـ **Digital I/O** وسننتقل إلي جزء آخر وهو **Analog I/P**