Design and Implementation of a DC Motor Control Platform Using the LabVIEW Programming Environment and the Microcontroller

Dr. Their Ahmad Ibrahim*
Ibrahim Tayseer Nwairah**

(Received 24 / 8 / 2023. Accepted 11 / 12 / 2023)

□ ABSTRACT □

The research deals with the design and implementation of a DC motor control platform intended for educational purposes using the Arduino microcontroller and the LabVIEW software environment. The designed system monitors the working status of the DC motor, measures its parameters (intensity of the current drawn - rotational speed - torque - load ratio) and draws its speed-torque curve, as well as an emergency stop when the nominal values of the current are exceeded. The results of testing the designed system on a parallel induction DC motor with a power of 24W showed that the designed system performs its functions correctly and achieves its desired purpose in terms of:

• The possibility of monitoring the values of the parameters of the tested motor in real time and the extent of changes in the values of these parameters when the load on the motor axis increases, as the value of the intensity of the current drawn increases, as well as the value of the torque, while the value of the rotational speed decreases.

• Controlling the start and stop of the motor and activating the emergency state when the nominal current value of the tested motor (2A) is exceeded in order to protect against overload.

• Plot the speed-torque curve of the tested motor at different load levels.

Keywords: Control, pressure sensor, DC motor, LabVIEW, Arduino uno.

Copyright **:**Tishreen University journal-Syria, The authors retain the copyright under a CC BY-NC-SA 04

* **Professor, Department of Industrial Automation, Faculty of Technical Engineering, Tartous University. Tartous, Syria.** thieribrahim@gmail.com

** **Postgraduate Student, (Master), Department of Control and Industrial Automation Faculty of Technical Engineering, Tartous University. Tartous, Syria.** IbrahimNwairah@tartous-univ.edu.sy

تصميم وتنفيذ منصة تحكم بمحرك تيار مستمر باستخدام البيئة البرمجية LabVIEW والمُتحكّم الصغري

د. ثائر أحمد إبراهيم •

إبراهيم تيسير نويره • •

(تاريخ الإيداع 24 / 8 / 2023. قُبِل للنشر في 11/ 12 / 2023)

ملخّص

يتناول البحث تصميم وتنفيذ منصة تحكم بمحرك تيار مستمر موجهة لأغراض تعليمية باستخدام المتحكم الصغري Arduino والبيئة البرمجية LabVIEW. حيث يقوم النظام المصمم بمراقبة حالة عمل محرك التيار المستمر وقياس البارامترات الخاصة به (شدة التيار المسحوب– سرعة الدوران– العزم –نسبة التحميل) ورسم المنحني سرعة– عزم الخاص به وكذلك الإيقاف الطارئ عند تجاوز القيم الاسمية للتيار. أظهرت نتائج اختبار النظام المصمم على محرك تيار مستمر ذو تحريض تفرعي واستطاعة 24W أن النظام المصمم يقوم بوظائفه بشكل صحيح ويحقق الغاية المطلوبة منه من حيث:

- إمكانية مراقبة قيم بارامترات المحرك المختبر في الزمن الحقيقي ومدى تغيرات قيم هذه البارامترات عند زيادة التحميل على محور المحرك حيث تزداد قيمة شدة التيار المسحوب وكذلك قيمة العزم الدوراني بينما تنقص قيمة سرعة الدوران.
- التحكم بتشغيل وإيقاف المحرك وتفعيل حالة الطوارئ عند تجاوز القيمة الاسمية للتيار الخاصة بالمحرك المختبر (2A) وذلك للحماية من التحميل الزائد.
- رسم منحني السرعة – العزم الخاص بالمحرك المختبر عند مستويات تحميل مختلفة.

الكلمات المفتاحية: تحكم، حساس ضغط، محركDC، LabVIEW،Arduino uno .

حقوق النشر : مجلة جامعة تشرين– سورية، يحتفظ المؤلفون بحقوق النشر بموجب الترخيص CC BY-NC-SA 04

• أستاذ، قسم هندسة الأتمتة الصناعية، كلية الهندسة التقنية، جامعة طرطوس، طرطوس، سورية thieribrahim@gmail.com

• • طالب دراسات عليا، قسم الأتمتة الصناعية، كلية الهندسة التقنية، جامعة طرطوس، طرطوس، سورية.
IbrahimNwairah@tartous-univ.edu.sy

مقدمة:

نتيجة الأزمة التي مرت بها البلاد تأثر الواقع التعليمي بشكل عام وبشكل خاص الناحية العملية، حيث تضررت المخابر في بعض المنشآت التعليمية وبعضها الأخر عانى من نقص بتجهيزات المخابر الحديثة نتيجة ظروف الحصار الاقتصادي والعلمي المفروض علينا. منها منظومات القياس الإلكترونية الحديثة التي تسهل تنفيذ التجارب العلمية للطلاب للمساعدة في الفهم والتعلم.

من هنا كانت الحاجة للاعتماد على الخبرة المحلية لسد النقص الحاصل في التجهيزات المخبرية التعليمية.

تم إجراء الكثير من الدراسات والأبحاث حول هذا الموضوع نذكر البعض منها. الدراسة [1] حيث قام فيها الباحث بتصميم وحدة تحكم تعمل في الزمن الحقيقي للتحكم بسرعة دوران محرك تيار مستمر باستخدام بيئة الـ MATLAB وتم التنفيذ عملياً باستخدام نظام تحصيل المعطيات DAQ مع البيئة البرمجية LabVIEW التي استخدمت لمراقبة دوران المحرك حيث تم استخدام حساس تيار مستمر لمراقبة التحميل الزائد ودارات الملائمة للتحكم بالتشغيل وإطفاء المحرك أظهرت النتائج أن جهاز التحكم المستخدم هو أكثر كفاءة من الطرق التقليدية وجيد للاستخدام في الزمن الحقيقي. وفي الدراسة [2] قام فيها الباحث بتصميم نظام للتحكم بسرعة دوران محرك تيار مستمر باستخدام البيئة البرمجية الـ LabVIEW والمتحكم Arduino تميز النموذج المصمم بالتكلفة المنخفضة وبساطة التصميم، تم استخدام مؤقت 555 ووحدات تحكم PID بيّن الباحث أنه تم التحكم بسرعة المحرك بشكل فعال من خلال التصميم المنفذ وأنه قابل للتوسع لأنواع مختلفة أخرى من المحركات مثل السيرفو. أما في الدراسة [3] فقد قام فيها الباحث بتقديم طريقة حديثة في قياس البيانات ومعالجتها وتخزينها بالاعتماد على حساسات LS200 والمحول A/D وقد أمكن من خلال واجهة رسومية مصممة باستخدام برنامج LABVIEW تحليل المخططات البيانية. في دراسة أخرى [4] فقد قدم فيها الباحث معلومات حول كيفية استخدام برنامج الـ Labview من أجل المراقبة وتجميع البيانات في الزمن الحقيقي واقترح البحث نظام ربط الـ Labview مع حساس الحرارة عبر لوحة Arduino للتحكم بتشغيل محرك تيار مستمر. وقد بين الباحث إمكانية استخدام هذا النظام لأتمتة أجهزة مختلفة مثل المراوح وأجهزة الإنذار بالحرارة العالية وإمكانية تطويره بحيث يرسل النظام رسالة sms عبر شبكة GSM إلى مركز المراقبة.

في هذا البحث سنقوم بتصميم نظام قياس الكتروني لمحرك تيار مستمر موجه لأغراض تعليمية باستخدام المتحكمات الصغرية الـ Arduino والبيئة البرمجية الـ LabVIEW وتطبيق نظام تحميل على محور محرك الـ DC بواسطة الكبح عن طريق محرك سيرفو بزوايا مختلفة وقياس نسبة التحميل عن طريق حساس ضغط FSR ومراقبة مدى تأثير نسبة التحميل على بارامترات المحرك (شدة تيار – سرعة دوران – عزم) ثم رسم منحني السرعة – عزم الخاص بمحرك التيار المستمر وتطبيق حالة الطوارئ عند تجاوز القيم الاسمية للتيار من خلال الواجهة التفاعلية المصممة ضمن البيئة البرمجية الـ LabVIEW.

أهمية البحث وأهدافه:

تكمن أهمية البحث في استثمار التقنيات الحديثة (البيئة البرمجية Labview) والمُتحكّمات (شرائح Arduino) في تصميم وتنفيذ منصة مخبريه موجهة للأغراض التعليمية تُمكّن من إجراء القياسات الإلكترونية لمحرك التيار المستمر بشكل مرن، مع تسجيل نتائج القياس وتحليلها. ورسم منحني السرعة –عزم الخاص بالمحرك. أما الهدف العام من هذا البحث هو التحكم بتشغيل المحرك وإيقافه عند تجاوز القيم الأسمية (حماية من التحميل الزائد).

طرائق البحث ومواده:

نقوم ببناء المخطط الصندوقي لنظام القياس والتحكّم والمراقبة المُقترح تصميمه وشرح عناصر ومكوّنات هذا النّظام. حيث تمّ اختيار المتُحكّم من نوع أردوينو (شريحة: Arduino uno) ومن أجل برمجته نستخدم البرنامج " Arduino IDE" المفتوح المصدر.

نستخدم حساس التيار من الطراز ACS71205B-T وحساس جهد مستمر VOLTAGE SENSOR25V وحساس لقياس سرعة الدوران (مشفر ضوئي) من طراز FC-03 وحساس الضغط FSR0.5 ومحرك تيار مستمر ذو تحريض تفرعي استطاعة 24W ومحرك سيرفو من طراز SERVO MG995، نصمم نظام تحميل باستخدام محرك السيرفو حيث يطبق إعاقة على محور المحرك بزوايا مختلفة وقياس قيمة التحميل باستخدام حساس الضغط.

من أجل تصميم الواجهة التفاعلية الخاصة بالمراقبة والتحكّم، نستخدم البيئة البرمجية LabVIEW التي تُعتبر مختبر افتراضي يضم معظم نُظم القياس الحديثة والتي تُمكّن من عرض وتحليل البيانات في الزمن الحقيقي.

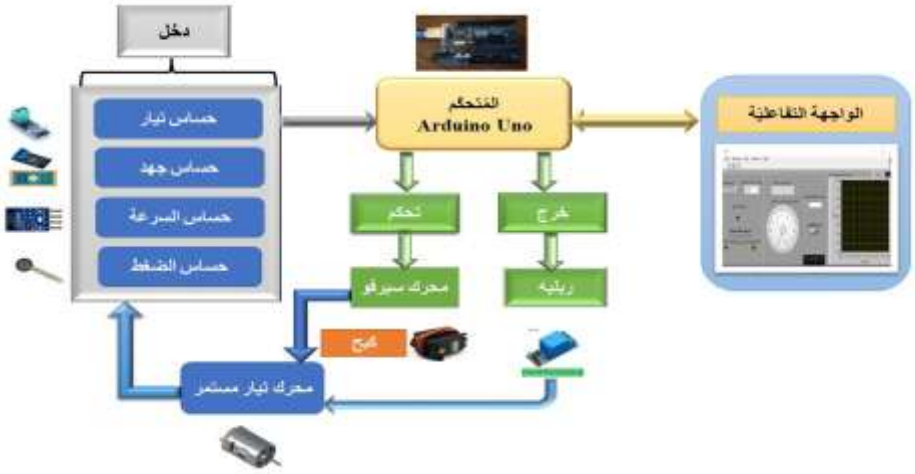
وفي النّهاية سنختبر النّظام المصمّم ونناقش النتائج.

1. تصميم نظام القياس والتحكّم والمراقبة الخاص بمحرك التيار المستمر:

يبيّن الشّكل (1) المخطط الصندوقي لنظام القياس والتحكّم والمراقبة المُقترح حيث استخدمنا أربع حسّاسات (الجهد والتيّار والضغط والسرعة) لقياس البارامترات الخاصة بالمحرك.

يقوم المُتحكّم بقراءة بيانات الحسّاسات ومُعالجتها ليتم إرسالها إلى الواجهة التفاعلية المصممة باستخدام البيئة البرمجية LabVIEW.

يتم التحكم بتشغيل وإيقاف المحرك وتحديد مستوى التحميل المطبق على هذا المحرك من خلال واجهة تفاعلية مصممة ضمن البيئة البرمجية الـ LABVIEW التي ترسل الأمر إلى المتحكم Arduino Uno عبر المنفذ التسلسلي USB الذي يتحكم بزاوية دوران محرك السيرفو المستخدم لتطبيق إعاقة على محور محرك الـ DC، تم استخدام حساس الضغط لقياس نسبة الضغط المطبقة على محور محرك الـ DC ليتم عرضها عن طريق المتحكم على الواجهة التفاعلية كنسبة تحميل وفي نفس الوقت يتم عرض قيمة شدة التيار والسرعة والعزم وكذلك منحني السرعة – عزم الخاص بالمحرك.



الشكل (1): المخطط الصّندوقي لنظام القياس والتحكّم والمراقبة المُقترح

1.1. مُتحكّم النّظام (Control Unit):

قمنا باختيار متحكّم الـ Arduino وذلك لرخص ثمنه وعدم الحاجة إلى جهد كبير لتأمين التغذية له واستهلاكه المنخفض للتيار الكهربائي. بالإضافة إلى سهولة التعامل معه وبساطة لغة البرمجة، حيث يتم برمجته عن طريق الحاسوب باستخدام البرنامج "Arduino IDE" المفتوح المصدر.

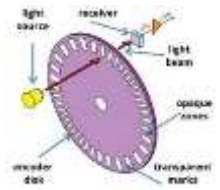
تتوفر شرائح الآردوينو بأنواع عديدة. تم اختيار المُتحكّم المُصغّر " Arduino uno " الشكل (2)، وذلك لأنه يلبي متطلبات النظام المراد تصممه بالإضافة للتّكلفة القليلة [5].

الشكل (2): شريحة Arduino UNO.

2.1. حسّاسات ومحركات النّظام (Sensors and Motors):

الجدول (1): الحساسات والمحركات المستخدمة في تصميم النظام

يبين الشكل (3) حساس الضغط، هو عبارة عن مقاومة حساسة للضغط، يمتلك منطقة استشعار للضغط حيث تتغير قيمة هذه المقاومة تبعا للضغط المطبق على منطقة الاستشعار حيث كلما زاد الضغط أو القوة المطبقة انخفضت قيمة المقاومة[6] .	 **الشكل (3): حساس الضغطFSR**
يقيس الحساس الموضّح في الشّكل (4) قيمة التّيار المستمر بمجال تتراوح قيمته من 0 إلى 5A. حساس التيار هو جهاز يكشف مرور التيار الكهربائي المار في السلك ويولد إشارة تتناسب مع هذا التيار، يمكن أن تكون الإشارة المتولدة جهدا تناظريا أو تياراً أو حتى خرجاً رقمياً [7].	 **الشكل (4): حساس التّيار المستمر ASC712-05B-T**
يقيس الحساس الموضّح في الشّكل (5) قيمة الجهد المستمر بمجال تتراوح قيمته من 0 إلى 25V [7].	 **الشكل (5): حساس الجهد المستمر**

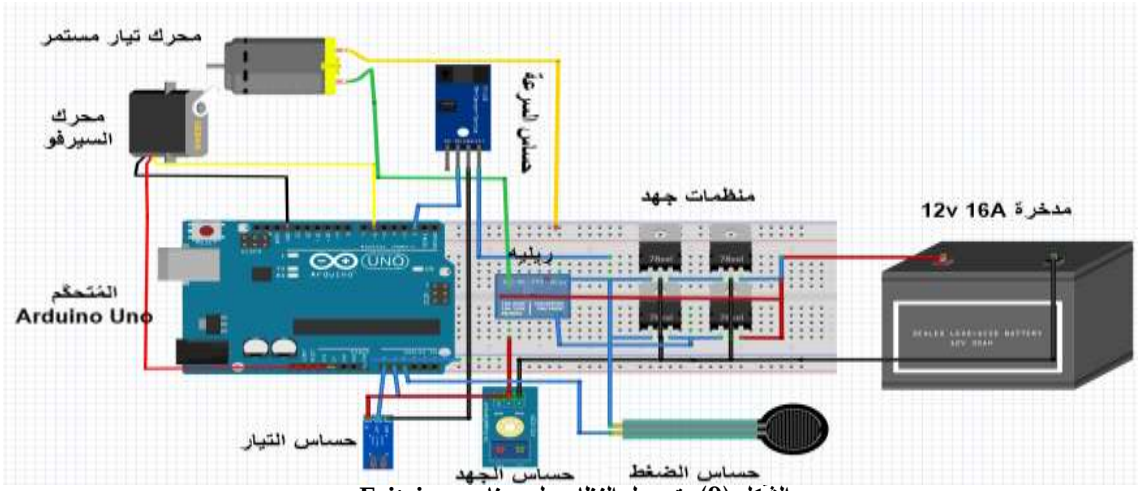
يقيس الحساس الموضّح في الشكل (6) قيمة سرعة دوران المحرك. الحساس ببساطة عبارة عن مصدر ضوء وحساس ضوء، يفصل بينهم قرص مثبت على محور الدوران في المحرك، والقرص يحتوي على عدد معين من الثقوب التي تسمح بمرور الضوء عبرها وعند دوران المحرك يمر الضوء بشكل متقطع من خلال القرص وبالتالي تنتج سلسلة من النبضات الكهربائية يمكن قراءتها من خلال حساس الضوء وبتردد 50HZ.	 **الشكل (6): حساس السرعة FC-03**

سنعرض فيما يلي وبشكل موجز الحسّاسات والمحركات اللّازمة لتصميم نظام القياس التحكّم والمراقبة المُقترح:

يوضح الشكل (7) محرك التيار المستمر ذو تحريض تفرعي المستخدم لاختبار النظام المصمم [8]. المواصفات الفنية للمحرك المستخدم: الجهد: 12v شدة التيار الاسمية: 2A الاستطاعة: 24w	**الشكل (7): محرك التيار مستمر**
يوضح الشكل (8) محرك السيرفو المستخدم لتطبيق الإعاقة على محور محرك التيار المستمر[9] . المواصفات الفنية لمحرك السيرفو المستخدم (servo mg995) عزم دوران 11 Kg/cm وجهد التشغيل 6V حتى 4.8 V	**الشكل (8) محرك السيرفو MG995**

3.1. التنفيذ الإلكتروني للنّظام(Electronic implementation of the system) :

قمنا بدايةً برسم مخطّط التّوصيل لمكوّنات النّظام باستخدام البرنامج Fritzing الموضح في الشّكل (9)، ومن ثم تم التّنفيذ العملي كما يبيّن الشّكل (10).



الشكل (9): توصيل النظام على برنامج Fritzing

الشّكل (10): تنفيذ دارة النّظام المصمّم

4.1. التّصميم البرمجي للنظام (Programmatic implementation of the system):

1.4.1. تصميم برنامج المُتحكّم (Design of the controller program):

يمكن برمجة شريحة الـ Arduino Uno عن طريق البرنامج Arduino IDE، تم اختيار البرنامج Arduino IDE، كونه مفتوح المصدر[5] .

في البداية قمنا بتضمين المكتبات الضرورية للعناصر المستخدمة في النظام، ثم عرّفنا أقطاب الدخل، ومن ثم تعريف كافة المتغيرات التي سيتم تخزين القيم فيها وارسالها إلى الواجهة التفاعلية. يبين الشكل (11) التعليمات الخاصة بتشغيل وإيقاف المحرك والتحكم بزاوية دوران محرك السيرفو المستخدم لتطبيق إعاقة على محور محرك الـ DC، حيث تم الحصول على قيم زوايا الدوران تجريبياً ووفق جدول استقبال المحارف من الواجهة التفاعلية في بيئة الـ LabVIEW حيث كل حالة تمثل مستوى تحميل.

```
void loop() {
  if ( Serial.available() ) {val = Serial.read();
    if ( val == 'd') {myservo.write(0);}
    if ( val == 'a') {myservo.write(33);}
    if ( val == 'b') {myservo.write(38);}
    if ( val == 'c') {myservo.write(43);}
    if ( val == 'e') {myservo.write(48);}
    if ( val == 's') {digitalWrite(7, LOW); MOTOR_ON = 1;}//on}
    if ( val == 'x') {digitalWrite(7, HIGH); MOTOR_ON = 0;}//off}
```

الشّكل (11): التعليمات الخاصة بتشغيل وإيقاف المحرك والتحكم بزاوية دوران محرك السيرفو

يبين الشكل (12) التعليمات البرمجية الخاصة بقراءة حساس الضغط عن طريق المتحكم وإظهار هذه القيمة على المؤشر في الواجهة التفاعلية بما يتناسب مع كل مستوى تحميل.

```
prusser = analogRead(A2);
prusser1 = map(prusser, 0, 245, 0, 10);
Serial.print('f'); Serial.println(prusser1);
```

الشّكل (12): التعليمات البرمجية الخاصة بقراءة حساس الضغط

يبين الشكل (13) التعليمات البرمجية الخاصة بقراءة حساس التيار عن طريق المتحكم وإظهار هذه القيمة على الواجهة التفاعلية عندما لا تتخطى 2A وإلاّ تستقبل برنامج الـ LabVIEW محرف خاص إظهار حالة الطوارئ لتجاوز التيار الاسمي وحماية المحرك من التحميل الزائد.

```
currentValue = CurrentRead();
Serial.print('j'); Serial.println(currentValue);
if (currentValue >= 2) {
  MOTOR_ON = 0;
  Serial.print('n'); digitalWrite(7, HIGH);
}
```

الشّكل (13): التعليمات البرمجية الخاصة بقراءة حساس التيار وتفعيل حالة الطوارئ.

يبين الشكل (14) التعليمات الخاصة بقراءة حساس السرعة وقيمة العزم المحسوبة عن طريق المتحكم وإظهار قيمة السرعة والعزم على المنحني البياني الذي يمثل علاقة السرعة مع العزم بما يتناسب مع كل مستوى تحميل.

```
if (MOTOR_ON == 1) {
  delay(10);
  rpss = spead();
  speed_rad = ((rpss * 2 * PI) / 60);
  Serial.print("z"); Serial.println(speed_rad);
  torque = power / speed_rad ;
  torque1 = torque * 10;
  Serial.print('v'); Serial.println(torque);}
```

الشّكل (14): التعليمات الخاصة بقراءة حساس السرعة وقيمة العزم المحسوبة

2.4.1. تصميم الواجهة التفاعلية للمراقبة والتحكّم ضمن البيئة البرمجية الـ LabVIEW:

يمكن من خلال البيئة البرمجية الـ LabVIEW التحكم وتحصيل المعطيات من مُتحكّم الآردوينو [10].

لربط الآردوينو مع الـ Labview نحن بحاجة للبرمجيات التالية:

1) Visa: Virtual Instrument Software Architecture وهي تعتبر معيار لتكوين أجهزة قياس وبرمجة والتي تتكون من واجهات للاتصالات التسلسلية أو واجهة أثرنيت أو USB فعند وصل شريحة الآردوينو مع الحاسب يقوم البرنامج السبق ذكره بإعلام الـ Labview بأن هناك وحدة أردوينو متصلة على منفذ معين ذو رقم معين. يتم إرسال عناصر التحكم (الإدخال / الإخراج) لتحديد الجهاز الذي نريد الاتصال به.

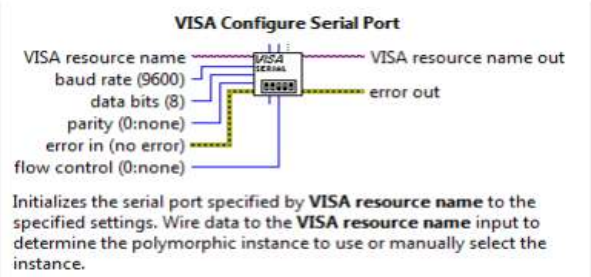
الجدول (2) يوضح الوظائف المختلفة التي يمكن استخدامها لإنجاز مهام متقدمة[11] .

الجدول (2): وظائف بلوكات معيار الـ Visa

البلوك	الوظيفة
Visa read	قراءة عدد بايتات محدد من الجهاز أو الواجهة المحددة عن طريق (visa resource name) وإعادة البيانات إلى المخزن المؤقت للقراءة.
Visa write	يكتب البيانات من المخزن المؤقت للكتابة إلى الجهاز المحدد عن طريق (visa resource name)

Case structure	يحتوي على حالات، يتم تنفيذ أحدها بالضبط عند تنفيذ البنية، تحدد القيمة المتصلة بمحدد الحالة المراد تنفيذها.
Build XY Graph Express VI	تنسيقات البيانات التي سيتم عرضها على الرسم البياني (x,y)
Visa close	لإغلاق جلسة عمل أو حدث محدد بواسطة visa resource name
Visa open	يفتح جلسة عمل على الجهاز المحدد بواسطة visa resource name ويعيد معرف جلسة العمل الذي يمكن استخدامه للاتصال بأي عمليات أخرى لهذا الجهاز.
visa read STB Function	يقرأ بايت حالة طلب الخدمة من الجهاز المستند إلى الرسائل المحدد بواسطة visa resource name
Time Delay Express VI	يُدخل تأخيرًا زمنيًا في قراءة قيم VI.

يتم تهيئة إعدادات الاتصال: عند فتح منفذ الاتصال التسلسلي تحدد visa المواصفات الافتراضية لإعدادات الاتصال حيث تكون كالتالي الشكل (15):



الشكل (15): تهيئة المنفذ التسلسلي

يمكننا تغيير هذه الإعدادات بما يناسب مجموعة بارامترات الاتصال حيث تشير الطرفية Visa resource name إلى الجهاز أو الأداة المطلوب من البرنامج التحكم بها والطرفيات (baud rate, data bit, parity, error in, flow control) إلى إعدادات بروتوكول الاتصال مع الجهاز المعرف من خلال Visa resource name وerror out تشير إلى الخطأ في الاتصال أو في الجهاز المعرف في حال حدوثه.

2) VI Package Manager:

وهو برنامج يتم تنصيبه عادة مع برنامج LabView نجده على موقع National Instruments. وهو متاح بشكل مجاني ويحتوي على مكتبات كثيرة خاصة ببرنامج LabView. ونحن نحتاجه هنا لتنصيب مكتبة LabView Interface for Arduino.

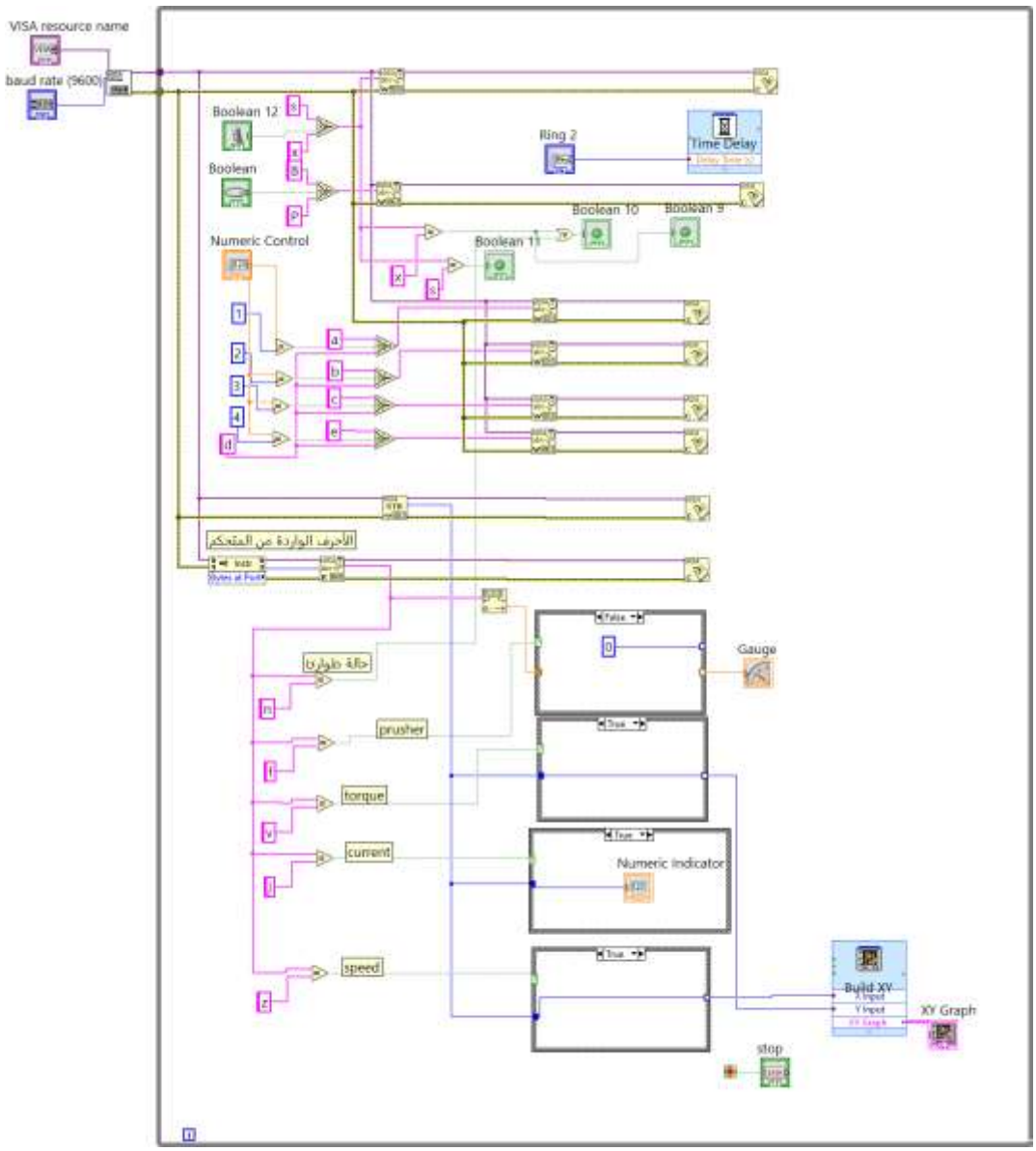
يوضح الشكل (16) واجهة برنامج التحكم والمراقبة والقياس للنظام المصمم، يحتوي البرنامج على أمر تشغيل وإيقاف المحرك، ومراقبة حالة المحرك (المحرك يعمل – متوقف– حالة طوارئ)، تحديد مستوى تحميل المحرك وعرض قيمة التحميل المقاس عن طريق حساس الضغط، عرض قيمة حساس التيار والعزم وسرعة الدوران عند كل مستوى تحميل، ورسم منحني السرعة–عزم.

الشكل (16) واجهة برنامج التحكم والمراقبة والقياس للنظام المصمم

من أجل تنفيذ الأوامر يجب أولاً تهيئة المنفذ التسلسلي بما يطابق بارامترات الاتصال التسلسلي التي تمت برمجة المتحكم (الآردوينو) عليها، يتم إرسال أمر التحكم عبر بلوك visa write الذي يقوم بكتابة الأمر إلى الجهاز الذي قمنا بتعريفه بواسطة visa configure serial port، يتم استقبال معطيات الحساسات عبر بلوك visa read الذي يقوم بقراءة المعطيات الواردة من الجهاز الذي قمنا بتعريفه كما يظهر في الشكل (17)، ويتم تنفيذ أوامر القراءة والكتابة وفق الجدول (3).

الجدول (3): يوضح أوامر القراءة والكتابة للواجهة التفاعلية المصممة باستخدام الـ Labview

حالة الكتابة (إرسال أمر تحكم إلى المتحكم)	حالة القراءة (استقبال معطيات عن طريق المتحكم)
تشغيل المحرك:S	حالة طوارئ (تحميل زائد فوق التيار الاسمي للمحرك):N
إطفاء المحرك:X	استقبال قيمة حساس الضغط:F
تحديد المستوى الاول في تحميل المحرك:A	استقبال قيمة حساس السرعة:Z
تحديد المستوى الثاني في تحميل المحرك:B	استقبال قيمة حساس التيار:J
تحديد المستوى الثالث في تحميل المحرك:C	استقبال قيمة العزم المحسوبة:V
تحديد المستوى الرابع في تحميل المحرك:E	
العمل على فراغ (حالة عدم تحميل):D	



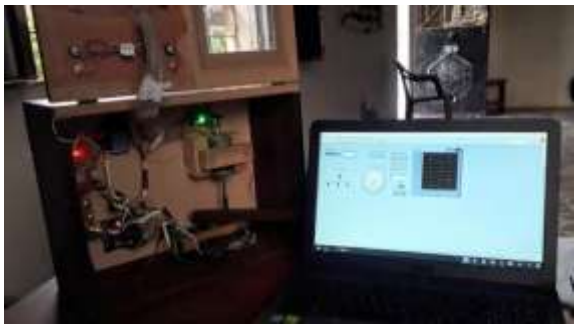
الشكل (17): المخطط الصندوقي لبرنامج التحكم والمراقبة والقياس لمحرك الـ DC بواسطة الـ LabVIEW

النتائج والمناقشة:

2. اختبار النّظام ومناقشة النّتائج:

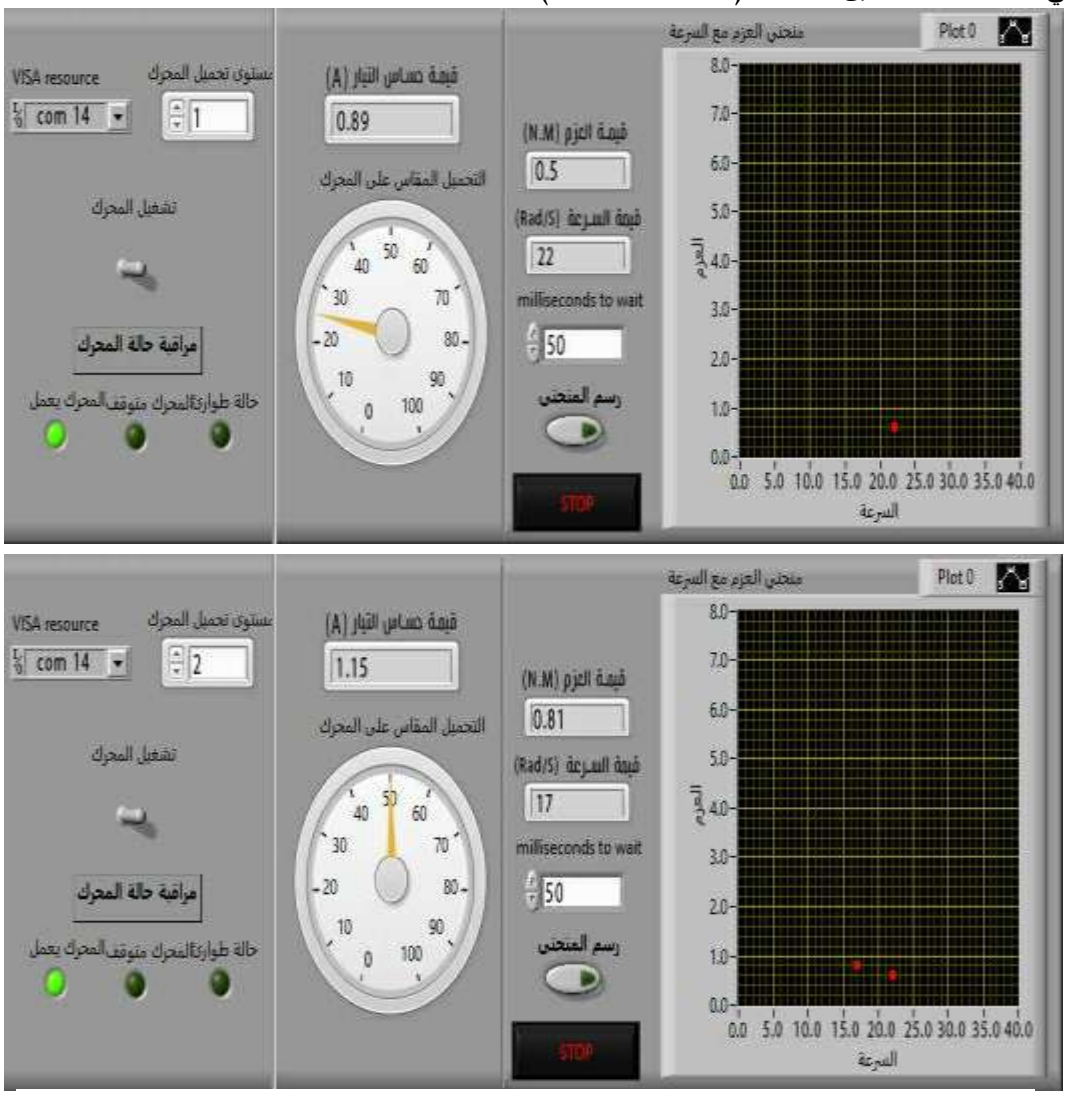
1.2 اختبار النظام:

تم توصيل التغذية المناسبة لجميع عناصر النظام المصمم عن طريق مدخرة 12V-16Aومجموعة منظمات جهد (حتى نضمن التشغيل في حال انقطاع التيار الكهربائي) واختباره على محرك تيار مستمر DC ذو تحريض تفرعي استطاعة 24W وتيار اسمي 2A لمعرفة البارامترات الخاصة به ومراقبتها والتحكم به كما يبين الشكل (18):



الشّكل (18): اختبار النّظام المُصمّم على محرك تيار مستمر DC ذو استطاعة 24 واط

تمّت قراءة البارامترات الخاصة بالمحرك من تيّار وسرعة دوران وعزم وكذلك قيمة التحميل المقاسة على المحرك عند أربع مستويات للتحميل، ومراقبة حالة المحرك (طوارئ – متوقف – عمل)، وكذلك رسم المنحني الخاص فيه (عزم – سرعة) وفق القيم المقروءة، تم تحديد تأخير زمني بين فترات أخذ القيم 50ms. تم عرض هذه البيانات على الواجهة التفاعلية المصممة ضمن البيئة البرمجية Labview عند أربع مستويات للتحميل وجهاز التحكم موصول على المنفذ التسلسلي COM14 كما تبين الأشكال (19–20–21–22):



الشّكل (20): عرض قيم البارامترات ومنحني سرعة – عزم لمحرّك الـ DC عند مستوى التحميل الثاني

الشّكل (21): عرض قيم البارامترات ومنحني سرعة – عزم لمحرّك الـ DC عند مستوى التحميل الثالث

الشّكل (22): عرض قيم البارامترات ومنحني سرعة – عزم لمحرّك الـ DC عند مستوى التحميل الرابع

نلاحظ من الأشكال (19–20–21–22) أنّ كلما انتقلنا من مستوى تحميل إلى أخر (زيادة التحميل على محور المحرك المستمر)

- تزداد قيمة شدة التيار المسحوبة (1.9A, 1.52A, 1.15A, 0.89A).
- يزداد العزم وتنخفض سرعة دوران المحرك كما هو مبين بالمنحني المرسوم بالحالة الرابعة من مستوى تحميل المحرك.

(22rad/s&0.5N.m ,17rad/s&0. 81N.m ,12rad/s&1. 58N.m ,6rad/s&3.8N.m)

ومن أجل اختبار حالة الطوارئ الخاصة بالمحرّك عند تجاوز قيمة التيار الاسمي 2A(حماية من التحميل الزائد) تم تشغيل ريليه التّوقف (الطّوارئ) بشكل فوري عند تجاوز القيمة المسموحة، وتمّ تشغيل ضوء على الواجهة التفاعلية للتعبير عن حالة التّوقف، يبيّن الشّكل (23) الإيقاف الآلي عند الوصول إلى حالة الطوارئ.

كذلك يمكن للمُستخدم إيقاف المحرّك عن طريق الضّغط على المفتاح "stop " في حال حدوث أي طارئ أخر (مثلاً ارتفاع درجة الحرارة لسبب ما) في الواجهة التفاعلية ضمن البيئة البرمجية الـ Labview، كما يوضّح الشّكل (23).

الشكل (23): تشغيل الإيقاف الآلي عند الوصول إلى حالة الطوارئ من الواجهة التفاعلية

2.2.مناقشة النتائج:

بيّنت نتائج الاختبار لنظام القياس والتحكّم والمراقبة الذي قمنا بتصميمه وتنفيذه أنّه يقوم بوظائفه ويحقق الغاية المطلوبة منه حيث تبين ما يلي:

- وفّر النّظام لنا إمكانية مراقبة قيم بارامترات محرك تيار مستمر ذو تحريض تفرعي (سرعة دوران، تيّار، عزم)، في الزّمن الحقيقي لمحركات ذات استطاعة تصل إلى 125 واط.
- وفر النظام رسم منحني السرعة – عزم الخاص بمحرك التيار مستمر ذو تحريض تفرعي.
- يفي المتحكّم الذي تم اختياره بالغرض من حيث السّرعة في تنفيذ التّعليمات ومُعالجة البيانات.
- تمكّن النّظام من تحديد حالة العمل الطّبيعي وحالة التوقف بالإضافة إلى حالة الإيقاف عند الطوارئ.
- تكلفة تنفيذ النّظام قليلة لا تتعدّى الـ 51$، كما يبين الجدول (4).

الجدول (4): حساب تكلفة النّظام

اسم المادّة	السّعر بالدولار	الكميّة	المجموع
متحكّم Arduino Uno	10	1	10
حسّاس الجهد	3	1	3
حسّاس التيّار	6	1	6
حسّاس الضغط	4	1	4
حساس سرعة	4	1	4
وحدة مخارج(ريليه)	3	1	3

محرك تيار مستمر 24W	5	1	5
محرك سيرفو mg995	11	1	11
توصيلات وملحقات	5	1	5
المجموع الإجمالي			51$

الاستنتاجات والتوصيات:

الاستنتاجات:

_ يمتاز نظام القياس والتحكّم والمراقبة المُصمّم بقلة تكلفته واستجابته السريعة لاعتماده على متحكّمات رخيصة الثمن ومتوفرة (Arduino Uno) وتفي بالغرض من حيث السرعة في تنفيذ التعليمات ومعالجة البيانات.

_ مكّنت البيئة البرمجية LabVIEW من تصميم واجهة تفاعلية لمراقبة البارامترات الخاصة بمحركات التيار المستمر في الزمن الحقيقي بالإضافة إلى التحكّم بها (حالة التحميل الزائد) ورسم المنحني سرعة – عزم الخاص بها.

التوصيات:

_ نوصي باستخدام الحساسات اللاسلكية لإرسال البيانات المتحسسة لاسلكيّاً إلى المتحكّم لتكريس فهم مبدأ الاتصال اللاسلكي.

_ نوصي باستخدام المُعالج المُصغّر الراسبيري باي (Raspberry Pi) كونه أقوى برمجياً في حال زيادة تعقيد النظام والحاجة إلى معالجة معقّدَة للبيانات.

References:

[1] SHAMIL, H. and MOHAMMED, A., "*Real time speed control of DC motor in Labview*", 7th International Conference on Electrical and Electronics Engineering, Antalya, Turkey, 2020.

[2] ANGALAESWARI, S.; KUMAR, A. and KUMAR, D., "*Speed control of permanent magnet (PM)DC motor using Arduino and LabVIEW"*, IEEE International Conference on Computational Intelligence and Computing Research, Chennai, India, 2016, pp.1-6.

[3] CARDONA, A.; CHICA, P. and BARRAGAN, O.," *BIPVS Basics for Design, Sizing, Monitoring, and Power Quality Measurement and Assessment*", In: Building-Integrated Photovoltaic Systems, Springer, Cham 2018, pp.17-33.

[4] VAIBHAV, M.; DAVANDE, C. and DHANAWADE, B., "*Real time temperature monitoring using Labview and Arduino*", International Journal of innovative Research in computer and communication engineering India, Vol. (4), No. (3), March 2016, 3409-3415.

[5] LOUIS, L.," *Working principle of Arduino and using it as a tool for study and research*", International Journal of Control, Automation, Communication and Systems India, vol. (1), No. (2), April 2016, 21-29.

[6] Nageshwar, N.; Krishnaa, S.G.; Narasimhan, S.L. and Venkatesan, M., "*Thrust Measurement Using Force Sensitive Resistor"*, IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 2017, pp. 1-4.

[7] THUMMAR, R.; CHUDASAMA, K.; KOSHIYA, R. and ODEDARA, D., "*Monitoring of Electric industrial process parameters using LabVIEW*", International journal of current engineering and technology Anand, Gujarat, India, Vol. (6), No. (6), Dec 2016, 2126-2129.

[8] Shanmuga, B.; Mythile, A.; Pavithra, S. and Nivetha, N.," *Parameter identification of a DC motor*", International Journal of Scientific & Technology Research Rohini Delhi, Vol. (9), N. (2), February 2020, 2277-8616.

[9] RAJAN, M.; SUBADEVI, S. and MADANRAJ, K., "*Application of Servo Motor Strategy to Avoid the Wastage of Consumable Energy*", international journal of creative research thoughts India, Vol. (8), No. (4), April 2020, 2320-2882.

[10] UGURLU, Y. and NAGANO, T., "*Project-based learning using LabVIEW and embedded hardware*", IEEE/SICE International Symposium on System Integration, Kyoto, Japan, 2011, pp. 561-566.

[11] Parmar, S.; Chudasama, K.; Vankar, J. and Gohil, A., "*Monitoring and Control of Speed of DC Motor using LabVIEW Environment*", International Journal for Research in Engineering Application & Management Gujarat, India, Vol. (3), No. (12), Mar 2018, 2454-9150.