ملحق (د)

نبذة عن استخدامات المايكروكنترولر في القياسات (لوحة أردوينو مثالًا)

١- مقدمة

انتشر في العصر الحاضر استخدام الكمبيوتر والمايكروكنترولر (Microcontroller) أو **المتحكم الصغير** في حياتنا اليومية، وأصبح من الأساسيات، سواء في البيت أو العمل أو حالة السفر. بل إن استخدام المايكروكنتولر في القياس والتحكم يكاد لا يحصى بما في ذلك الاستخدامات السلمية والحربية.

المايكروكنترولر هو مجموعة كبيرة من الدوائر المتكاملة (IC=Integrated Circuit) تشكل كمبيوتر صغير، ويحتوي على ذاكرة ومداخل ومخارج (Input & outputs) ويمكن إيصاله عادة بمصدر تيار مستمر وبرمجته، بحيث يمكن المستعمل من إدخال إشارات، سواء رقمية أو تماثلية، وإجراء ما يلزم عليها، ثم إظهارها إذا لزم الأمر على مخرجات مختلفة، مثل فتح أو غلق مفتاح على سبيل المثال، أو إظهار القيم على شاشة صغيرة. ولذلك يمكن أن يستخدم المايكروكنترولر مع أجهزة الاستشعار المختلفة لقياس بعض المتغيرات، مثل درجة الحرارة والتغير في الفولتية أو المقاومة أو التيار، ومن ثم حفظها على الكمبيوتر أو اتخاذ إجراء عليها. ويهدف هذا الملحق إلى تطبيق مباشر وسهل لاستخدام المايكروكنترولر في القياس والتحكم، وخصوصا لطلاب مادة قياسات.

وفي ما يلي بعض تطبيقات المايكروكنترولر:

١- الجوالات

٢- مشغل السي دي والـ دي في دي (CD/DVD)

٣- تشغيل والتحكم في الأدوات المنزلية الكهربائية (مثل: الغسالات، والأفران، والمايكروويف ... الخ)

٤- أجهزة الإنذار

٥- التحكم في الإضاءة

٦- البيوت الذكية

٧- التحكم في تشغيل المواتير

٨- الروبوت

٩- أجهزة القياس (درجة الحرارة، والرطوبة، والقوة ... الخ)

١٠- الألعاب

١١- الاعتناء بالصحة

١٢- الدفاع والحروب والأسلحة والطيران

١٣- الاستدامة وترشيد استخدام الطاقة

المايكروكنترولر ما هو إلا كمبيوتر في شكل مضغوط في شريحة دوائر متكاملة [١-٧]. فهو يحتوي على جميع العناصر الأساسية الموجودة في الكمبيوتر، مثل:

١- المايكروبروسوسير (Microprocessor)

٢- ذاكرة عشوائية (RAM)

٣- ذاكرة قراءة فقط (ROM)

٤- مخارج ومداخل (Inputs and Outputs)

٥- الهزاز أو الساعة (Oscillator)

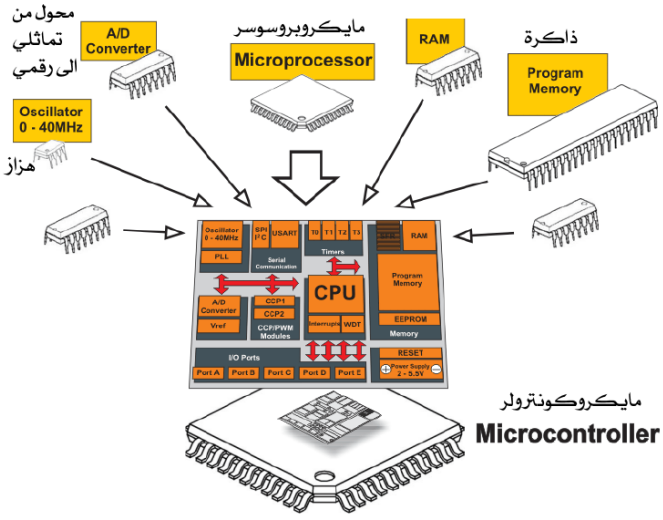
٦- انتربتر (Interrupter)

٧- مؤقتات (Timers)

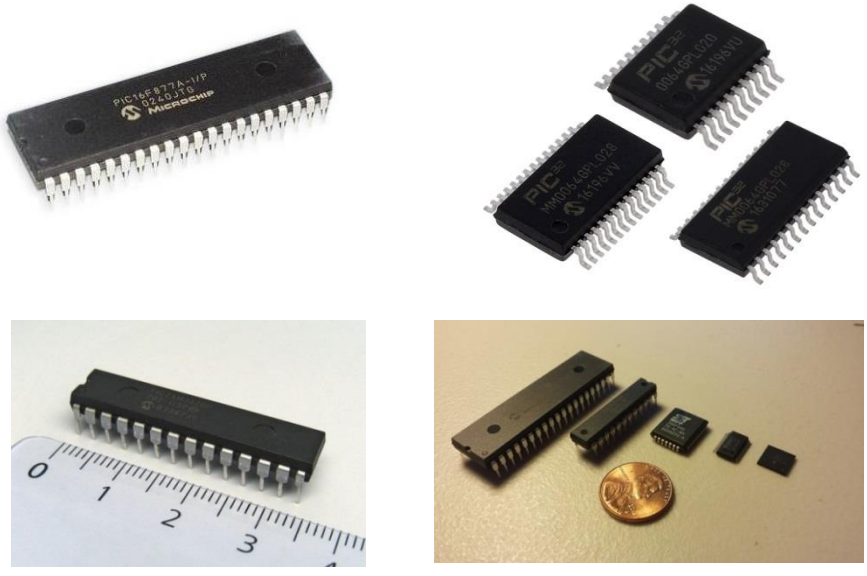
يبين الشكل (د ١) الأجزاء الأساسية في المايكروكنترولر [٤]. ويحتوي المايكروكنترولر على منافذ دخول وخروج تسمح للعالم الخارجي التعامل والتواصل مع المتحكم الصغير. يمكن أن تكون هذه المنافذ رقمية (مثل قراءة مفتاح مغلق أومفتوح)، أو تماثلية (مثل قراءة درجة الحرارة، أو الرطوبة، أو الفولتية)، وكذلك إصدار أوامر (مثل فتح أو إغلاق دائرة كهربائية).

وظهر في الآونة الأخيرة تطبيقات المايكروكنترولر في تخصص دراسي هندسي يسمى ميكاترونكس (Mechatronics) [٨،٩]، كخليط من مواد دراسية في تخصص الهندسة الميكانيكية ومواد دراسية في تخصص الهندسة الكهربائية والإلكترونية.

وهناك أشكال مختلفة للمتحكم الصغير، فبعضها يكون على شكل مستطيل وأخرى مربع الشكل. وهناك بعض التطبيقات لمايكروكنترولر بحجم الإبهام. ويتميز المتحكم الصغير بأن له أرجل (أو أطراف) كل طرف له مهمة معينة حسب تصميم المصنع. يوضح الشكل (د ٢) أشكالًا مختلفة من المايكروكنترولر، وتوضيح لحجمها الفعلي بمقارنتها بعملة معدنية صغيرة.



شكل (د ١) الأجزاء الرئيسية لمكونات المايكروكونتلور [٤].



شكل (د٢) بعض أشكال وأحجام المايكروكنترولر.

وهناك عدة شركات عالمية [١٠-١٧] لتصنيع المياكروكنترولر، منها على سبيل المثال القائمة الموجودة في جدول (د١). ولبرمجة المايكروكنترولر ومعرفة خصائصه، وجب على المبرمج الرجوع لمواصفات المايكروكنترولر حسب نوع المتحكم الصغير. وتوجد مواقع على الإنترنت [١٨-١٩] تعطي ما يسمى بصفحة المعلومات (DataSheet) لكثير من أنواع المايكروكنترولر، والمجسات، والقطع الإكترونية المختلفة. ويتم عادة برمجة المايكروكنترولر بإحدى اللغات (مثل Basic, C, Assembly)، ثم يتم حرق (burn) أو تنزيل البرنامج على المايكروكنترولر باستخدام وصلة بين الكمبيوتر والمايكروكنترولر تسمى بالمبرمج (Programmer) [١-٧]، ثم يتم توصيل أطراف (أو أرجل) المايكروكنترولر حسب الدائرة الإلكترونية/الكهربائية للمصمم، وبالطبع يتم توصيل المايكروكنترولر بمغذي تيار مستمر في حدود ٥-١٢ فولت، وكذلك وجب توفير هزاز (Oscillator) أو ساعة ضبط بسرعة تتناسب مع تصميم المايكروكنترولر (٤ ، أو ٨، أو ١٦ ميغاهرتز، مثلا). وتجدر الإشارة بوجود برامج كمبيوترية تحاكي وتماثل عمل المايكروكنترولر، ويمكن الاستفادة منها في عمل الدائرة على برنامج المحاكاة، وتشغيلها على الكمبيوتر، والتاكد من صحة التوصيلات والبرنامج قبل البدء الفعلي في عمل الدائرة، ومن أشهر هذه البرامج برنامج بروتيوس Proteus [٢٠]. وتوفر بعض الشركات العاملة في مجال المايكروكنترولر بعض لوحات التجارب والحقائب (kits) للطلاب وصاحبي الهويات [٥، ٢١،٢٢].

جدول (د١) أسماء ومواقع لبعض مصنعي المايكوكنترولر.

	الشركة	**الموقع على الإنترنت**
1	microchip	http://www.microchip.com/
2	Atmel	http://www.atmel.com/products/microcontrollers/avr/
3	Texas Instrument	http://www.ti.com/
4	Renesas	https://www.renesas.com/en-eu/
5	STMicroelectronics	http://www.st.com/en/microcontrollers.html
6	Intel	http://www.intel.com/content/www/us/en/embedded/products/quark/overview.html
7	Toshiba	http://toshiba.semicon-storage.com/us/product/microcomputer.html
8	NEC	http://www.nec.com/en/global/solutions/index.html

٢- لوحة أردوينو

قام مجموعة من المهتمين ببرمجة المياكروكنترولر في إيطاليا بتصميم لوحة أردوينو التي هي أصلًا متحكم صغير (مايكروكنترولر) في لوحة بها مصدر كهربائي ووحدة اتصال مع الكمبيوتر عام ٢٠٠٥م (Massimo Branzi, David Cuartielles, Gianluca Martino) [٢٣-٢٤]، وذلك بغرض تسهيل برمجة المتحكم الصغير للهواة وغيرهم.

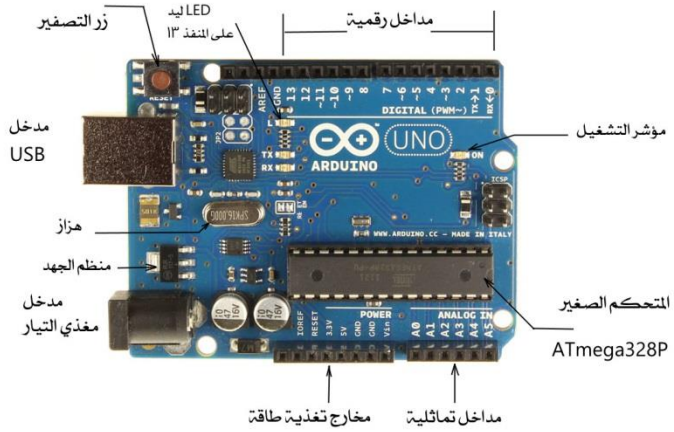
تتكون لوحة أردوينو التي تبلغ أبعادها حوالي ٧ سم طول في ٥ سم عرض، كما يظهر الشكل (د٣) الأجزاء الرئيسية الآتية:

١- مايكروكونترولر من نوع Atmel ATmega328P
٢- عدد ١٤ مدخل/مخرج رقمي (من صفر إلى ١٣) منها عدد ٦ مداخل متغير النبضة (Pulse Width Modulating =PWM)
٣- عدد ٦ مداخل /مخارج تماثلية (Analog) (A0 إلى A5)
٤- مدخل لـ USB للتوصيل بالكمبيوتر
٥- منافذ للطاقة الكهربائية (٥ أو ٣,٣ فولت يمكن للمصمم استخدامها في دائرته الإكترونية) وأرضي (Ground).
٦- مدخل مغذي كهربائي (٧-١٢ فولت) تيار مستمر لتشغيل اللوحة
٧- هزاز بسرعة ١٦ ميجاهرتز
٨- زر إعادة التشغيل
٩- ICSP للبرمجة على التوازي
١٠- محول إشارة تماثلية الى رقمية بقدرة ١٠ بتات (١٠٢٤ مستوى)

ومن خصائص المايكروكنترولر Atmel ATmega328P المستخدم على لوحة أردوينو ما يلي:

السرعة	١٦ ميغاهيرتز	ملاحظات
الذاكرة	٣٢ كيلوبايت	
ذاكرة SRAM	٢ كيلوبايت	تسجيل المتغيرات بصورة مؤقتة
ذاكرة Flash	٢٩ كيلوبايت	مساحة تخزينية للبرنامج بشكل دائم
ذاكرة EEPPROM	١ كيلوبايت	ذاكرة تحتفظ بقيمتها بعد فصل التيار
ذاكرة جزء البرنامج	٢ كيلوبايت	ذاكرة بها جزء برمجي تم تخزينه في

(Bootloader)	المتحكم الصغير مسبقا، وهو المسؤل عن استقبال البرامج من الكمبيوتر وتخزينها في المتحكم



شكل (د٣) الأجزاء الرئيسية للوحة أردوينو.

بعض مميزات لوحة الأردوينو

١- لوحة أردوينو بمكوناتها مفتوحة المصدر (أي أن معلوماتها وتوصيلاتها متوفرة) لمن أراد أن ينتجها ويستخدمها

٢- برمجة أردوينو أيضًا مفتوحة المصدر

٣- سهولة البرمجة إلى حد كبير

٤- رخص الثمن

٥- توجد تطبيقات كثيرة لا محدودة ومتاحة للكل على شبكة الإنترنت

٦- يوجد كثير من الكتب والدروس (الكثير منها مجاني) والمواقع لشرح استخدامات وتطبيقات لوحة الأردوينو

٧- يمكن ربط الأردوينو ببرامج أخرى معروفة مثل MATLAB و JAVA

٨- توفر كثير من الملحقات والمستشعرات (المجسات) الجاهزة للاستخدام

٩- توفر أنواع مختلفة من لوحة أردوينو حسب الاستخدام (صغير جدا، وصغير، ومتوسط الحجم، وكبير)

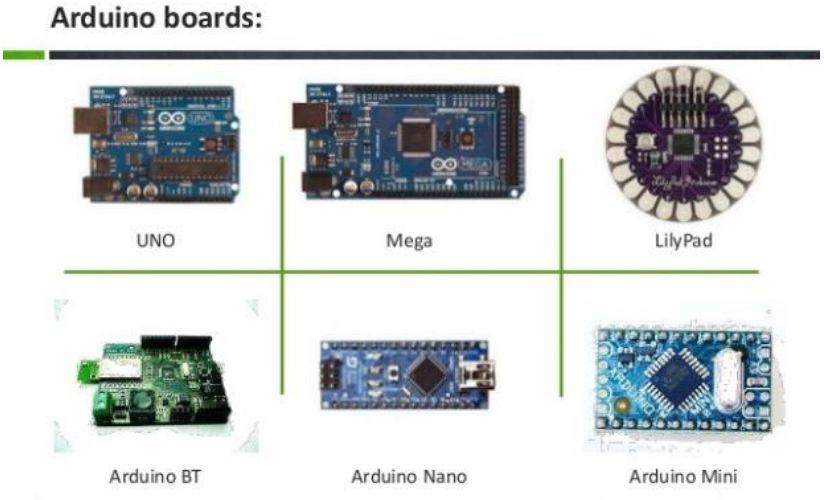
١٠- لغة برمجة الأردوينو هي لغة سي (C) مطورة وإلى حد ما سهلة التعلم والاستخدام

أنواع لوح الأردوينو

هناك أنواع مختلفة من لوح الأردوينو حسب حاجة المستخدم، منها الصغير ومنها الكبير، وتختلف في خصائصها وعدد المنافذ، وللمستخدم اختيار النوع المناسب له. ومن ضمن أنواع لوح الأردوينو:

١- أردوينو أونو Uno وهي الأكثر انتشارا واستخداما
٢- أردوينو ميغا Mega
٣- أردوينو نانو Nano
٤- أردوينو ميني Mini
٥- أردوينو ليلي باد LiLypad
٦- أردوينو Demvlive
٧- أردوينو Boarduino

يبين الشكل (د٤) بعض أنواع لوحات أردوينو.



شكل (د٤) بعض أنواع لوحات أردوينو.

باستخدام أحد لوحات الأردوينو يمكن عمل كثير من المشاريع حسب الحاجة. وفيما يلي بعض المشاريع التي يمكن تنفيذها باستخدام لوحة الأردوينو أونو (Arduino – uno)، وهي اللوحة التي سيتم عرض المشاريع عليها في هذا الملحق:

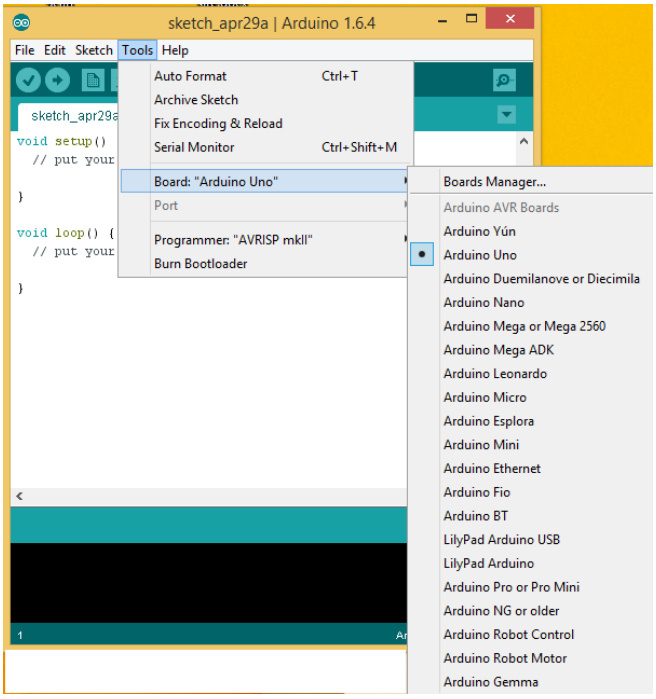
١- قياس بعض المتغيرات، مثل: درجة الحرارة، والضغط، والرطوبة، والمسافة، و تدفق الموائع ... الخ، وإظهار قيمها على شاشة العرض الكرستال LCD
٢- استخدامات التحكم في الروبوت
٣- التحكم في المخارج، سواء رقميًا أو تماثليًا، مثل إطفاء الإضاءة أو غلق مفتاح
٤- التحكم في تشغيل المواتير
٥- أجهزة الإنذار

٣- برمجة الأردوينو

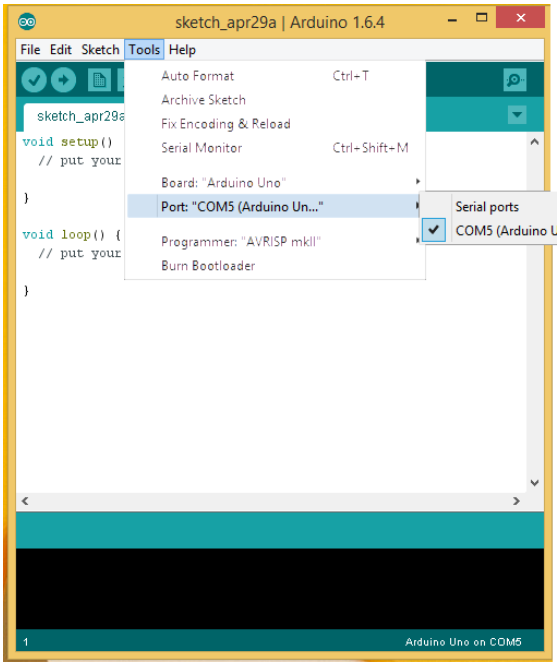
للبدء في استخدام لوحة الأردوينو، يحتاج المستخدم لتنزيل برنامج على الكمبيوتر من موقع (https://www.arduino.cc) [٢٥]، وهذا البرنامج هو برنامج مفتوح المصدر، ويمكن تثبيته على جهاز يعمل على نظام وندوز (Windows) أو لينكس (Linux). ويعرف بأنه نظام البيئة المتكاملة (IDE-Integrated Development Environment). بعد تثبيته على جهاز الكمبيوتر يمكن المستخدم من برمجة لوحة الأردوينو عبر سلك (يو أس بي USB)، كما يمكن تنزيل مكتبات متنوعة لتشغيل مجسات وأجهزة أخرى. وبعد تثبيت برنامج البيئة المتكاملة ينبغي التاكد من أمرين قبل البدء في استخدام وبرمجة الأردوينو، وهما:

١- اختيار نوع لوحة أردوينو المشبوكة مع الكمبيوتر (هنا يتم اختيار Arduino-uno) كما يظهر في الشكل (د ٥).

٢- التاكد من توصيل لوحة أردوينو على أحد منافذ التسلسل (انظر الشكل د٦).



شكل (د٥) واجهة برنامج البيئة المتكاملة IDE لبرمجة الأردوينو واختيار لوحة أردوينو.



شكل (د٦) اختيار توصيل لوحة أردوينو بأحد منافذ التسلسل (COM).

يتكون أي برنامج للاردبونو من جزئين رئيسين هما:

١- الجزء الخاص بالتعريفات (Setup) ويقوم الأردوينو بالمرور عليه وتنفيذه **مرة واحدة فقط**. وهو يحتوي على بعض التعريفات المبدئية، مثل المخارج والمداخل وأي عملية يطُلب تنفيذها مرة واحدة فقط. ويكتب عادة كالاتي:

```
setup()
{
// Here you insert command to be processed once
}
```

حيث إن الأوامر بين القوسين الهلاليين { } يتم تنفيذها مرة واحدة فقط. مع العلم أن الخطين المائلين في بداية السطر يمثلان إشارة إلى الأردوينو أن هذا السطر يحتوي على معلومات للمبرمج (Comment statement) ولا تؤخذ في اعتبار الأردوينو عند تنفيذ البرنامج. وإذا أراد المبرمج أن يضيف أكثر من سطر كتعريفات (Comment statements) فعليه كتابتها بين خط مائل ونجمة كالاتي:

```
/* These lines are comment statements
and are not to affect the Arduino program
One can write program description for example*/
```

٢- والجزء الثاني من برنامج أردوينو خاص بالعمليات المتكررة (Loop)، وهذه العمليات يقوم الأردوينو حسب البرمجة **بإعادة تكرارها** بصورة مستمرة حسب طلب المبرمج، فعلى سبيل المثال يمكن للمبرمج أن يكرر إرسال إشارة رقمية إلى أحد المنافذ. وأيضًا تكون هذه الأوامر محددة بقوسين هلالين { }.

إضافة إلى الجزئين الرئيسيين، ربما يحتاج المبرمج إضافة مكتبة (Library) بها بعض التعريفات والأوامر التي تسهل في استخدام القطع المستخدمة في المشروع، مثل استخدام شاشة العرض البلوري (الكرستال) السائل (Liquid Crystal Display LCD) لإظهار النتائج. وكذلك قد يحتاج المبرمج لإضافة بعض التعريفات أو الملاحظات للمتغيرات، سواء كانت أرقام أو حروف أو غيرها، ويوضع هذا الجزء مباشرة بعد تعريف المكتبات. ويظهر الشكل (د٧) أجزاء برنامج الأردوينو الرئيسية. ويتوفر على شبكة المعلومات العالمية الإنترنت كثير من الفيدوهات والكتب والدروس والشروح لتعليم استخدام وبرمجة الأردوينو [٢٦- ٣١].

بعد كتابة البرنامج يمكن التاكد من صحة البرنامج وعدم وجود أخطاء بضغط زر ، كما يمكن إرسال البرنامج إلى لوحة أردوينو باستخدام زر ، انظر الشكل (د٧). ولمزيد من المعلومات يمكن الرجوع لأحد المراجع السابقة أو الفديوهات على شبكة الإنترنت.

شكل (د٧) المكونات الرئيسية لبرمجة الأردوينو.

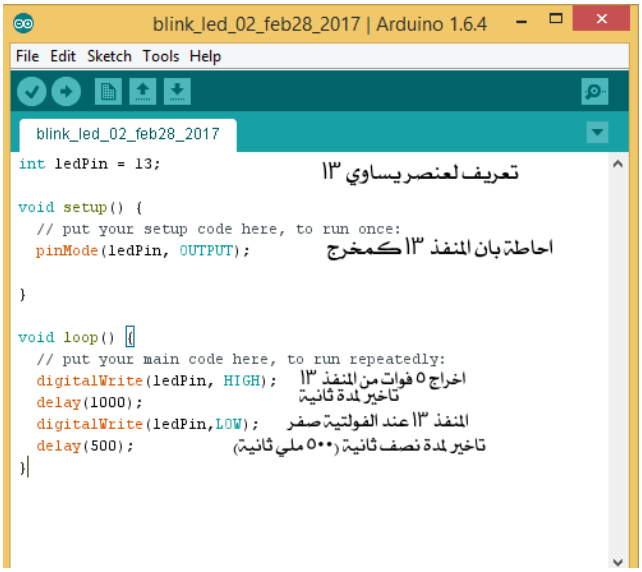
٤- نماذج من استخدام لوحة الأردوينو في القياس والتحكم

في هذا الجزء، سيتم استعراض بعض المشاريع البسيطة لاستخدامات لوحة الأردوينو في القياس والتحكم. وسيشتمل الجزء على عرض ميزات لوحة الأردوينو في التعامل مع المنافذ، سواء كانت رقمية أو تماثلية، بالإضافة إلى عرض نتائج القياس، سواء على المخرج المتسلسل على شاشة الكمبيوتر أو على شاشة الكرستال.

٤,١ التحكم في تشغيل لمبة دايود

هذا من أبسط برامج البداية للتعرف على الأردوينو. يوجد على لوحة الأردوينو لمبة دايود (LED) على المخرج ١٣. يمكن كتابة برنامج للأردوينو للتحكم في إضاءة اللمبة لفترة زمنية محددة بإرسال جهد ٥ فولت من أحد المنافذ الرقمية. ويبين الشكل (د٨) البرنامج، حيث يتم تعريف المنفذ ١٣ كمخرج في جزء التعريفات. تم في السطر الأول تعريف المتغير ledPin يساوي ١٣. وفي جزء الإعدادات لمرة واحدة تم تعريف المنفذ ١٣ على أساس أنه مخرج باستخدام الأمر pinMode(ledPin,OUTPUT). وفي جزء الأوامر المتكررة تتم إضاءة لمبة الدايود باستخدام الأمر digitalWrite(ledPin,HIGH) والثبات على ذلك لمدة ١٠٠٠ ملي ثانية (أي ثانية واحدة) باستخدام

الأمر delay(1000)، ثم يتم إرسال الأمر digitalWrite(ledPin,LOW) لإيقاف إضاءة اللمبة لمدة نصف ثانية باستخدام الأمر delay(500).

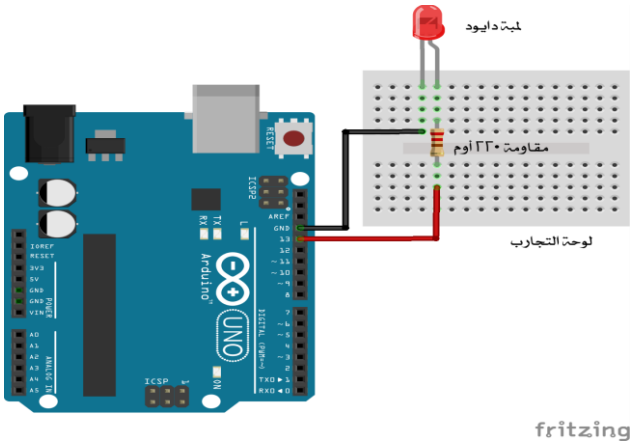


شكل (د٨) برنامج إضاءة لمبة الدايود على المنفذ ١٣.

ويوضح الشكل (د٩) موقع لمبة الدايود الموجودة على لوحة الأردوينو. وحسب البرمجة السابقة، فإن هذه اللمبة ستضيء لمدة ثانية وتنطفيء لمدة نصف ثانية، وتتكرر العملية بشكل مستمر. لاحظ من خلال برنامج الأردوينو الموضح في الشكل (د٩) أن أوامر الأردوينو تكتب باللون الأحمر وأنها يجب أن تكتب بدقة، سواء كانت الحروف كبيرة أو صغيرة، وأن كل أمر ينتهي بنقطة وفاصلة أي ;).

شكل (د٩) موقع لمبة الدايود على لوحة الاريدنو على المنفذ رقم ١٣.

ويمكن تركيب لمبة دايود خارجية على منفذ ١٣ مع ضرورة وضع مقاومة كهربائية بقيمة حوالي ٣٠٠ أوم لوضع حد لشدة التيار المار بالدايود الضوئى، كما يوضح الشكل (د١٠). لاحظ أنه تم استخدام لوحة التجارب (BreadBoard) لتثبيت الدائرة الإكترونية. وفي هذه الحالة عندما يتم استخدام نفس البرنامج السابق، فإن اللمبة التي على لوحة الأردوينو واللمبة الخارجية ستضيئان وتنطفئان معًا حسب المدة الزمنية الموجودة في البرنامج.



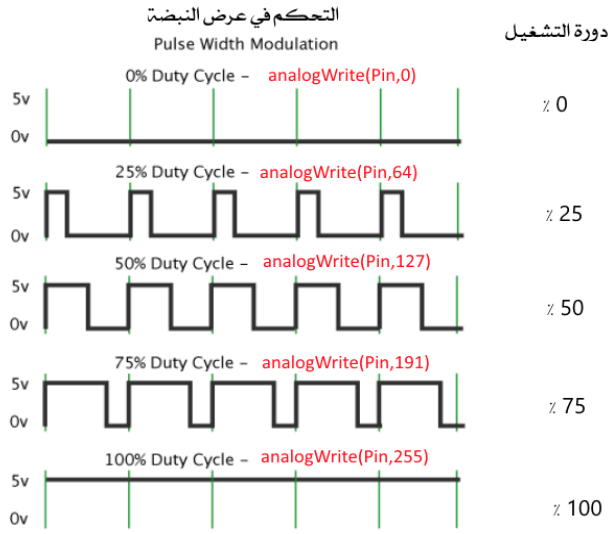
شكل (د١٠) التحكم في لمبة دايود خارجية مثبتة على المنفذ الرقمي رقم ١٣.

٤,٢ التحكم في إضاءة الدايود الضوئي باستخدام التحكم في عرض النبضة (PMW=Pulse Width Modulating)

يوجد على لوحة الأردوينو عدد ١٤ منفذًا رقميًا، وهي من صفر إلى ١٣. ومن ضمن هذه المنافذ تم تعريف المنافذ (٣، ٥، ٦، ٩، ١٠، ١١) كمنافذ تعطي إشارة رقمية، ولكي يتم التحكم في عرض النبضة. حيث إن الأمر analogWrite(Pin,duty) يقوم بتشغيل المنفذ (Pin) حسب الرقم (duty). حيث duty يمثل دورة التشغيل، وهو رقم بين ٠ إلى ٢٥٥. وإذا كان هذا الرقم بصفر، فإن الخارج من المنفذ يكون (صفر) فولت. وإذا كان هذا الرقم بـ ٢٥٥ فإن الخارج من المنفذ يكون (٥) فولت.

يوضح الشكل (د١١) كيفية التحكم في دورة التشغيل على أحد المنافذ (Pin). أما الشكل (د١٢) فيعرض برمجة الأردوينو للتحكم في تشغيل لمبة الدايود توهجا وانطفاءً. ويوضح الشكل (د١٣) الدائرة الكهربائية لتوصيل الأردوينو (منفذ رقم ٥) مع لوحة التجارب (BreadBoard)، مع ضرورة

وضع مقاومة بمقدار حوالي ٣٠٠ أوم على التسلسل مع لمبة الدايود، لوضع حد لشدة التيار المار بلمبة الدايود، وإذا لم يتم ذلك فإنه من المحتمل بسب شدة التيار أن تتعطل اللمبة.



شكل (د ١١) التحكم في دورة التشغيل على أحد منافذ التي تتحكم في عرض النبضة (PWM).

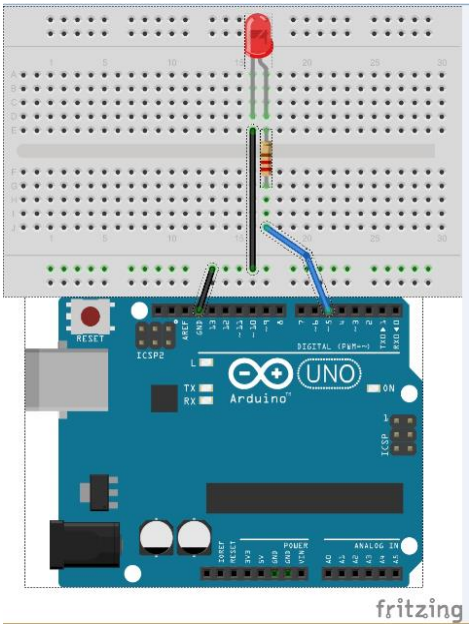
تعريف المنفذ المتصل بلمبة الدايود

```
int Pin=5;
int i;
void setup() {
  // put your setup code here, to run once:
pinMode(Pin,OUTPUT);
Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  for (i=1;i<255;i++)
  {
    analogWrite(Pin,i);
    delay(50);
    Serial.println(i);
  }
   // digitalWrite(Pin,HIGH);
delay(2000);
//digitalWrite(Pin,LOW);
 for (i=255 ; i>1; i--)
  {
    analogWrite(Pin,i);
    delay(50);
    Serial.println(i);
  }
delay(1000);
}
```

يتم زيادة توهج اللمبة حسب قيمة المتغير i تصاعديا

يتم التقليل في توهج اللمبة حسب قيمة المتغير i تنازليا

شكل (د ١٢) برنامج التحكم في شدة إضاءة لمبة دايود مثبتة على منفذ ٥.

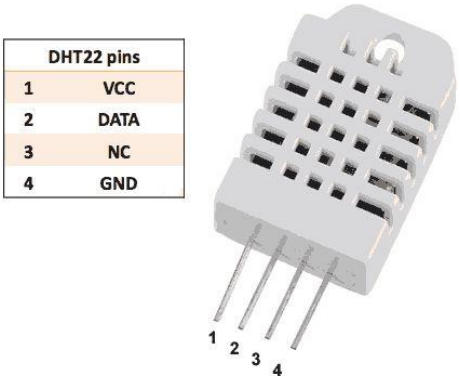


شكل (د ١٣) توصيل لمبة الدايود على المنفذ رقم ٥ والتحكم في وهجها بالأردوينو.

٤,٣ قياس درجة الحرارة والرطوبة النسبية وإظهارهما على المخرج المتسلسل

هناك أنواع مختلفة من المجسات التي تستخدم في قياس درجة الحرارة، ويمكن للوحة الأردوينو التعامل معها [٣٢]، مثل الازدواج الحراري، والثيرمستر، والدوائر المتكاملة IC. وفي هذا المشروع سيتم استخدام مجس صغير بحجم الإبهام هو DHT22 لقياس درجة الحرارة والرطوبة النسبية [٣٣]، كما يظهر في الشكل (د١٤). لهذا المجس أربعة أطراف (هي: مغذي التيار VCC، ونتائج القياس DATA، والأرضي GND، وطرف غير مستخدم). يبين الجدول (د٢) بعض خواص هذا المجس، وتوجد مكتبة [٣٤] لاستخدامها مع هذا المجس تسمى dht.h ستتم إضافتها للبرنامج. سيتم في هذا المثال اظهار نتائج قياس درجة الحرارة و الرطوبة على شاشة الكمبيوتر (أو ما يسمى بالمخرج المتسلسل Serial output).

يبين الشكل (د١٥) كيفية توصيل المجس من نوع DHT22 مع لوحة الأردوينو، لاحظ أن معلومات القياس الرقمية موصلة على المنفذ رقم ٢ على لوحة الأردوينو. والجدير بالذكر أن الدوائر الإلكترونية التي تبين توصيلات القطع مع لوحة الأردوينو، مثل الشكل (د١٥) وما يشابه، قد تم رسمها ببرنامج مميز لعمل الدوائر الإكترونية هو Fritzing [٣٥]، ويمكن الحصول عليه من الإنترنت واستعماله مجانا.

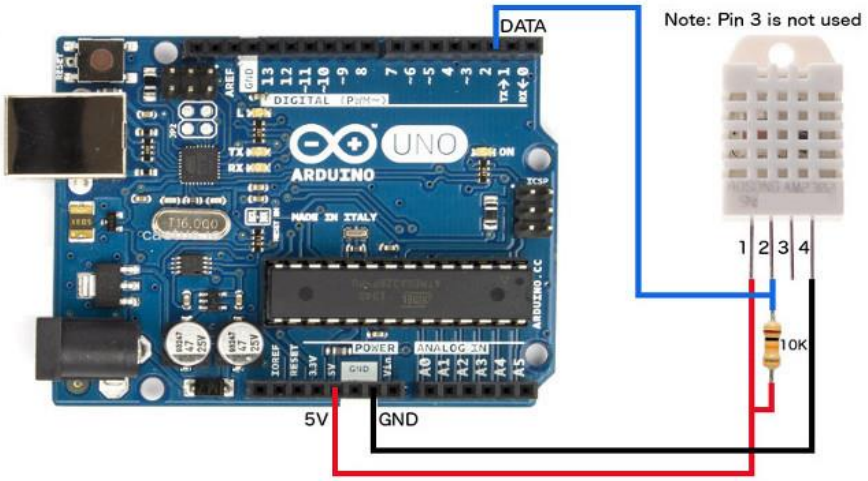


شكل (د١٤) مجس لقياس درجة الحرارة والرطوبة من نوع DHT22.

جدول (د٢) بعض خواص مجس درجة الحرارة والرطوبة النسبية DHT22 [٣٣].

التيار المغذي	٣,٣ إلى ٦ فولت تيار مستمر
الإشارة الخارجة من المجس	رقمية
نوع المجس	مكثف بوليمر Polymer capacitor
مدى درجة الحرارة	-٤٠ إلى ٨٠ درجة مئوية
مدى الرطوبة النسبية	من صفر إلى ١٠٠٪
صحة القراءة درجة الحرارة	أقل من -+ ٠,٥ درجة مئوية
صحة القراءة للرطوبة النسبية	-+ ٢٪ وأقصى خطأ ٥٪
الدقة/ التقسيم	٠,١ درجة مئوية و ١٪ للرطوبة النسبية
الأبعاد	١٤*١٨*٥,٥ مم

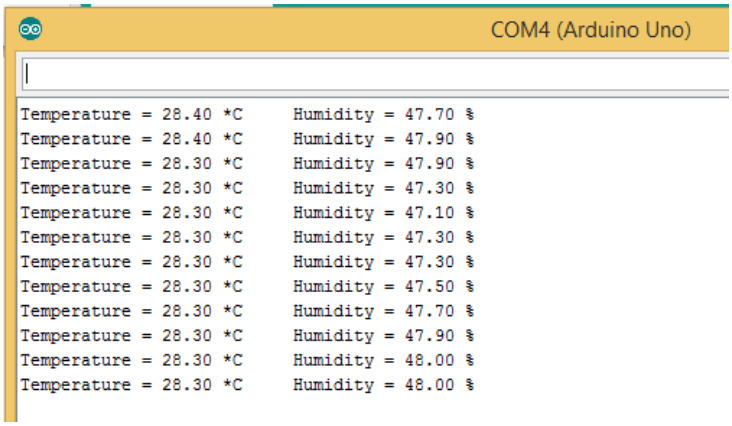
أما الشكل (د١٦) فهو نسخة من برنامج الأردوينو لقياس درجة الحرارة والرطوبة النسبية بهذا المجس، وتم في هذه البرمجة إعطاء أمر بإخراج النتائج على شاشة المخرج المتسلسل Serial output على شاشة الكمبيوتر، كما يظهر في الشكل (د١٧). لاحظ أنه تمت إضافة مكتبة خاصة بهذا المجس وهي dht.h ، كما هو موضح في رأس البرنامج.



شكل (د١٥) توصيل مجس درجة الحرارة والرطوبة DHT22 على لوحة الأردوينو.

```
#include <dht.h>      اضف المكتبة الخاص بمجس قياس درجة الحرارة و الرطوبة
/*  DHT11/ DHT22 Sensor Temperature and Humidity Tutorial

 * You can find the DHT Library from Arduino official website
 */

#define dataPin 2 //  يتم القياس عن طرق المدخل الرقمي ٢
dht DHT; //                    يتم احداث كائن DHT

void setup() {
  Serial.begin(9600);   ابدأ او نشط المخرج المتسلسل
}
void loop() {
  int readData = DHT.read22(dataPin); // أقرأ القياسات من المدخل ٢
  float t = DHT.temperature; //  أقرأ درجة الحرارة
  float h = DHT.humidity; //    أقرأ الرطوبة النسبية

  // Printing the results on the serial monitor
  Serial.print("Temperature = ");       اكتب النتائج على المخرج المتسلسل
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print("    Humidity = ");
  Serial.print(h);
  Serial.println(" % ");

  delay(2000); // Delays 2 secods, as the DHT22 sampling rate is 0.5Hz
}
```

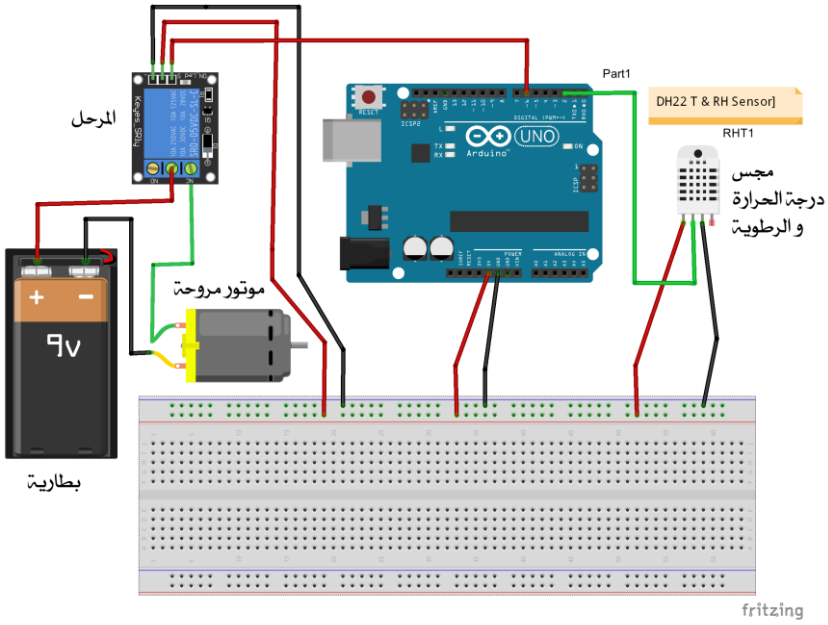
شكل (د١٦) برنامج أردوينو لقياس درجة الحرارة والرطوبة باستخدام DHT22 وإظهار القياس على المخرج المتسلسل.



شكل (د١٧) إظهار نتائج قياس درجة الحرارة والرطوبة باستخدام مجس DHT22 على المخرج المتسلسل.

يمكن إضافة عنصر تحكم للبرنامج السابق، بحيث يتم تشغيل مروحة (على سبيل المثال) إذا ما تجاوزت درجة الحرارة قيمة معينة، وذلك باستخدام مرحل (Relay). المرحل كما يظهر في شكل (د١٨) له جهتان. الجهة التي يتم توصيلها إلى لوحة الأردوينو، وهي عبارة عن ثلاثة أسلاك (التغذية الكهربائية ٥ فولت، والأرضي والإشارة الرقمية من الأردوينو)، وفي الجهة الثانية ثلاث توصيلات، وهي: عادة مغلق، وعادة مفتوح، والمشترك (Normally closed NC, Normally open NO, Common C) للتوصيل مع دائرة تشغيل المروحة، انظر الشكل (د١٨). تم تعديل البرنامج السابق لقياس درجة الحرارة، بحيث يتم تشغيل مروحة تعمل على جهد ٩ فولت عندما تكون درجة الحرارة أكثر من ٢٩ درجة مئوية، وتقفل إذا كانت درجة الحرارة أقل من ذلك. الشكل (د١٩) يبين الدائرة الإلكترونية والتوصيلات على الأردوينو. والشكل (د٢٠) يبين البرنامج بعد التعديل لعملية التحكم. لاحظ أنه تم استخدام المنفذ رقم ٦ للتحكم في تشغيل المرحل وتوصيل COM مع NC لتشغيل موتور المروحة حينما تكون درجة الحرارة أكثر من ٢٩ درجة مئوية.

شكل (د١٨) المرحل وتوصيلاته مع الأردوينو ودائرة تشغيل الموتور.



شكل (د ١٩) الدائرة الكهربائية الإلكترونية لمجس درجة الحرارة والأردوينو والمرحل والموتور وتوصيلاتها.

```
#include <dht.h>
/*  DHT11/ DHT22 Sensor Temperature and Humidity Tutorial
 * You can find the DHT Library from Arduino official website
 */
const int fanPin =6;  تعريف المنفذ الرقمي ٦ للتحكم في تشغيل ماتور المروحة
#define dataPin 2 // Defines pin number to which the sensor is connected
dht DHT; // Creats a DHT object

void setup() {
  Serial.begin(9600);
  pinMode(fanPin, OUTPUT);
}
void loop() {
  int readData = DHT.read22(dataPin); // Reads the data from the sensor
  float t = DHT.temperature; // Gets the values of the temperature
  float h = DHT.humidity; // Gets the values of the humidity

  // Printing the results on the serial monitor
  Serial.print("Temperature = ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print("    Humidity = ");
  Serial.print(h);
  Serial.println(" % ");
  if (t > 29)    digitalWrite(fanPin,HIGH);  الجزء الذي تم اضافته للتحكم في تشغيل المروحة
  else digitalWrite(fanPin,LOW);
    delay(2000); // Delays 2 secods, as the DHT22 sampling rate is 0.5Hz
}
```

شكل (د ٢٠) برنامج الأردوينو لمجس درجة الحرارة والرطوبة والتحكم في تشغيل الموتور حسب قيمة درجة الحرارة

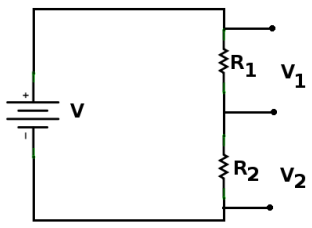
٤,٤ قياس الفولتية وإظهار قيمتها على شاشة LCD

سيتم في هذا المشروع توضيح ميزتين مهمتين للوحة الأردوينو، وهما إمكانية الأردوينو من استقبال إشارة تماثلية (فولت متغير في هذه الحالة) والشيء الآخر هو إمكانية توصيل شاشة كرستالة (LCD) على الأردوينو لإظهار النتائج المقاسة. من المعروف أن دائرة تقسيم الجهد (Voltage divider circuit) تقسم الجهد الكلي المغذي إلى جهدين على مقاومتين حسب قيمة المقاومتين. فلو كانت المقاومتين متساويتين يتم تقسيم الجهد بالتساوي، وإذا كانت القيمتان مختلفتين فإنه يتم التقسيم حسب العلاقة الآتية

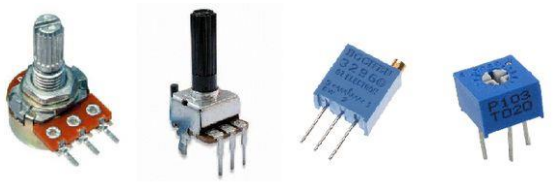
$$V_1 = V\left(\frac{R_1}{R_1+R_2}\right) \qquad \text{(D-1)}$$

حيث V_1 هو الجهد حول المقاومة الأولى R_1، و V هو الجهد المغذي للدائرة، انظر الشكل (د٢١). وسيتم استخدام مقاومة متغيرة (Variable resistance) بحث يتم التحكم في الجهد الخارج حسب تغيير المقاومة. يبين الشكل (د٢٢) بعض أنواع المقاومات المتغيرة. وسيتم استخدام مغذي الفولت الموجود في لوحة الأردوينو، وهو ٥ فولت. كما سيتم استخدام إحدى المنافذ التماثلية (A0, A1, A2, A3, A4, A5) لقياس الفولتية من دائرة تقسيم الجهد. وعلى لوحة الادينو نظام تحويل الإشارة من تماثلية إلى رقمية بسعة ١٠ بت وهو ما يماثل ١٠٢٤ قسما (لاحظ أن عدد الأقسام (Divisions) يساوي الرقم ٢ مرفوع لأس يساوي عدد البتات). أي لو كانت قيمة الجهد تساوي ٥ فولت فرقميا هي ١٠٢٤، ولو كانت مثلا تساوي ٥٠٠ رقميا فإن القيمة المقابلة لها هي ٢,٤٤ فولت وهكذا. لذلك سيتم إدخال الجهد كقيمة تماثلية على المنفذ التماثلي A0 وقياس القيمة الرقمية، وطالما أن عدد البتات هو ١٠ وأقصى جهد هو ٥ فولت، فيتم استخدام المعادلة الآتية لمعرفة قيمة الفولتية المقاسة. وإذا افترضنا أن القيمة المقاسة رقميا هي x فإن الفولت (أو الجهد) المقاس هو

$$V = \left(\frac{x}{1024}\right) * 5 \qquad \text{(D-2)}$$



شكل (د٢١) دائرة تقسيم الفولت.



شكل (د٢٢) أشكال مختلفة من المقاومات المتغيرة التي يتم التحكم في قيمتها إما يدويًا أو باستخدام مفك.

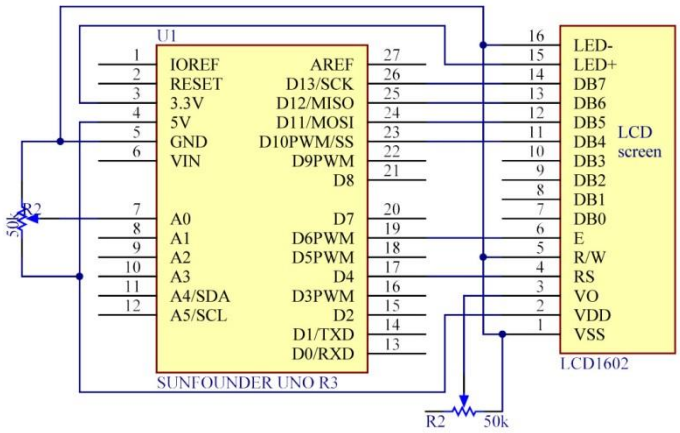
يظهر الشكل (د٢٣) برنامج الأردوينو لقياس الفولتية، علما بأنه تمت إضافة مكتبة شاشة الكريستال LCD، والقيمة المقاسة رقميا هي في المتغير val، وسيتم إظهار قيمة الفولت على شاشة الكرستال LCD [٢٩]. لاحظ أنه لتشغيل شاشة العرض LCD تم توصيل ستة منافذ رقمية من لوحة الأردوينو لتشغيل وإظهار النتائج على شاشة LCD. وفي هذا المثال، كما يظهر في السطر الرابع من البرنامج (انظر الشكل د٢٣) فقد تم توصيل المنافذ ٤، ٦ ، ١٠، ١١ ، ١٢، ١٣ من لوحة الأردوينو على المنافذ RS، E، DB4، DB5، DB6، DB7 من منافذ شاشة الكرستال LCD وذلك على التوالي.

```
#include <LiquidCrystal.h>          لاضافة مكتبة شاشة العرض
/*****************************************************/
//const int analogIn = A0;
LiquidCrystal lcd(4, 6, 10, 11, 12, 13); توصيل الشاشة مع الاردينو
float val = 0;                          تعريف المتغير و نوعه
/*****************************************************/
void setup()                            جزء التعريفات
{
  Serial.begin(9600);          تعريف المخرج المتسلسل على شاشة الكمبيوتر
  lcd.begin(16, 2);                    تعريف شاشة الكرستال
  lcd.print("Voltage Value:");         طباعة على شاشة الكرستال
}
/*****************************************************/
void loop()                             جزء الاوامر المتكررة
{
  val = analogRead(A0);
  val = 0.5*val + 0.5*analogRead(A0);        حسابات الفولت
  val = val/1024*5.0;
  Serial.print(val);
  Serial.print("V");                    طباعة على المخرج المتسلسل
  Serial.println();
  lcd.setCursor(6,1);
  lcd.print(val);                       طباعة على شاشة الكرستال
  lcd.print("V");                       ثانية تاخير
  delay(1000);
}
```

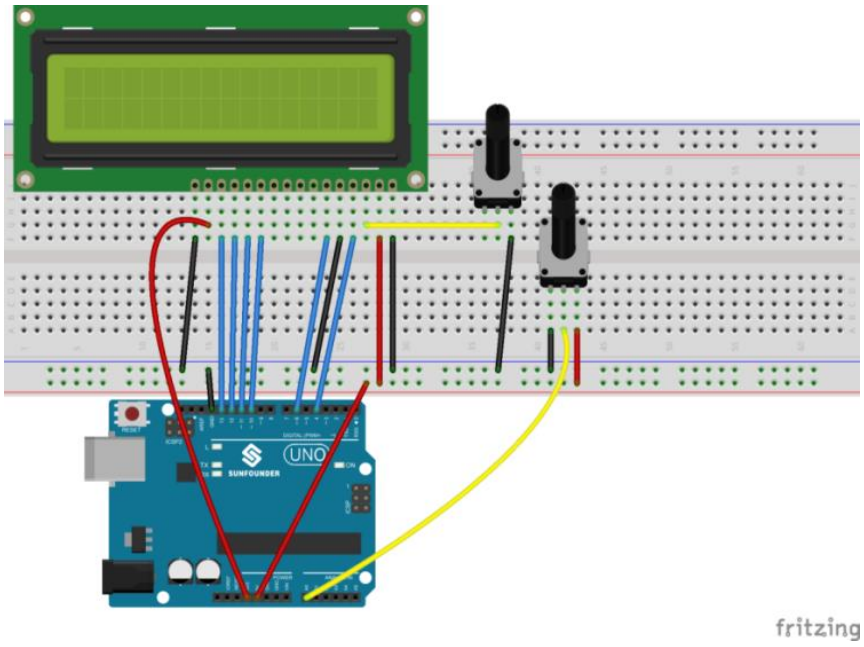
شكل (د٢٣) برنامج أردوينو لقياس الفولتية وإظهار قيمتها على شاشة كرستال.

ويبين الشكل (د٢٤) توصيل شاشة الكرستال (LCD) مع لوحة الأردوينو، ويمكن الرجوع لأحد مراجع الأردوينو لمعرفة تفاصيل أكثر عن توصيلات شاشة العرض وأوامرها [٣١-٢٨]. أما

الشكل (د٢٥) فيظهر توصيلات الدائرة الإلكترونية كاملة، وتم استخدام برنامج Fritzing [٣٥] لعملها. في الشكل (د٢٥) تظهر مقاومتين متغيرتين إحدهما تستخدم في التحكم في شدة إضاءة شاشة الكرستال LCD، والمقاومة المتغيرة الأخرى تستخدم للتحكم في مقدار الجهد المدخل على المنفذ التماثلي A0 للوحة الأردوينو، والذي يراد قياسه.



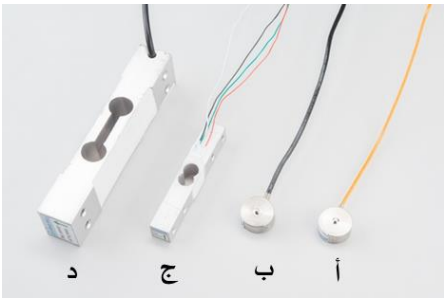
شكل (د٢٤) توصيل شاشة الكرستال LCD مع الأردوينو.



شكل (د٢٥) توصيل دائرة قياس الفولتية (الجهد) باستخدام لوحة الأردوينو.

٤,٥ قياس الوزن أو القوة باستخدام خلية الحمل

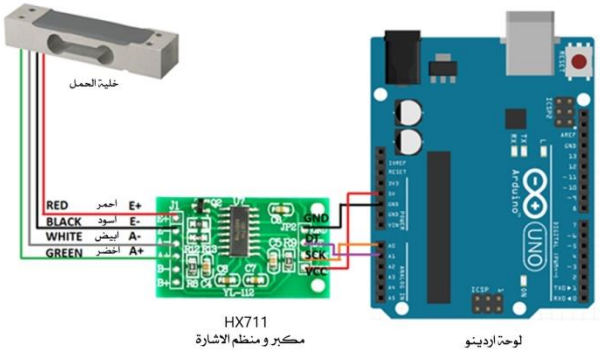
سبق الحديث عن استخدام قنطرة ويتستون مع خلية الحمل (load cell) لقياس الوزن أو القوة. في هذا المثال سيتم استخدام لوحة الأردوينو لقياس الوزن. هناك عدة أشكال لخلية الحمل، ويبين الشكل (د٢٦) بعضًا من أشكال خلايا الحمل المستخدمة. وفي هذا المثال سيتم استخدام النوع ج الذي يظهر في الشكل، والذي هو عبارة عن عتبة (beam) بها فتحتان صغيرتان مثبت عليهما أربعة أسلاك انفعال، وعن طريق الأربعة أسلاك ذات الألوان: أحمر، وأسود، وأبيض، وأخضر، يتم إيصال التيار المغذي، وقياس الإشارة من أسلاك الانفعال. ولأن الإشارة (signal) من خلية الحمل صغيرة جدًا، فإنه يتم استخدام مضخم للإشارة (signal amplifier)، حيث يقوم بتكبير الإشارة وجعلها محسوسة لدى لوحة الأردوينو. في حالتنا هذه، سيتم استخدام المضخم HX711 [٣٤]، كما سيتم معايرة الفولت المقاس من خلية الحمل بأوزان معيارية معروفة.



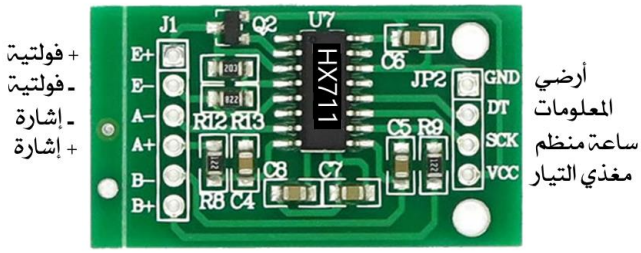
شكل (د٢٦) بعض أنواع خلايا الحمل المستخدمة في قياس الوزن والقوة [٣٠].

كما يوضح الشكل (د٢٧) أنه لخلية الحمل أربعة أسلاك هي: الأحمر، والأسود، والأبيض، والأخضر، حيث يتم استخدام السلكين الأحمر والأسود كتغذية كهربائية للخلية. والسلكان الأبيض والأخضر هما اللذان يجلبان الإشارة للمضخم HX711. وفي الجهة الأخرى من المضخم، يتم توصيل الأطراف الأخرى، وهي مغذي التيار VCC، والأرضي GRD، والإشارة DAT، وكذلك منظم الساعة CLK على لوحة الأردوينو، كما هو موضح في الشكل (د٢٧). وللتوضيح أكثر فإن طريقة توصيل المضخم HX711 مع قنطرة ويتستون موضحة في الشكل (د٢٨).

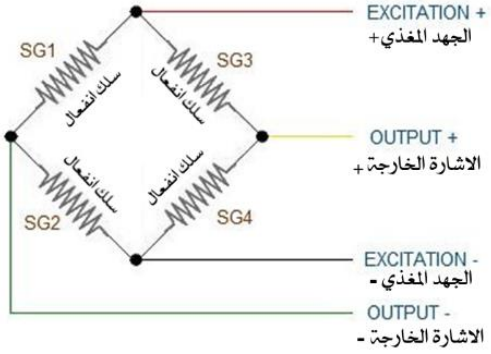
يوضح الشكل (د٢٩) قنطرة ويتستون والأربع مقاومات (أسلاك الانفعال) والإشارة الداخلة (الجهد المغذي) والإشارة الخارجة (signal).



شكل (د٢٧) توصيل خلية الحمل بلوحة الأردوينو باستخدام مضخم الإشارة HX711 لقياس الوزن.



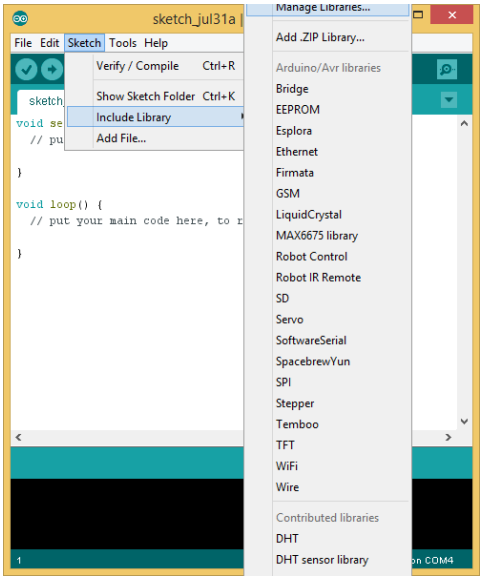
شكل (د٢٨) تفاصيل توصيلات مضخم الإشارة HX711.



شكل (د٢٩) توصيل أسلاك الانفعال على خلية الحمل مع تيار التغذية وقياس الإشارة الخارجة.

ينبغي لمصمم عملية قياس الوزن باستخدام خلية الحمل باستخدام المضخم الرجوع إلى معلومات المصمم للمضخم HX711، وذلك بالرجوع إلى ورقة المعلومات (Data sheet) من أحد مواقع معلومات القطع الإكترونية مثل [١٨، ١٩، ٣٤].

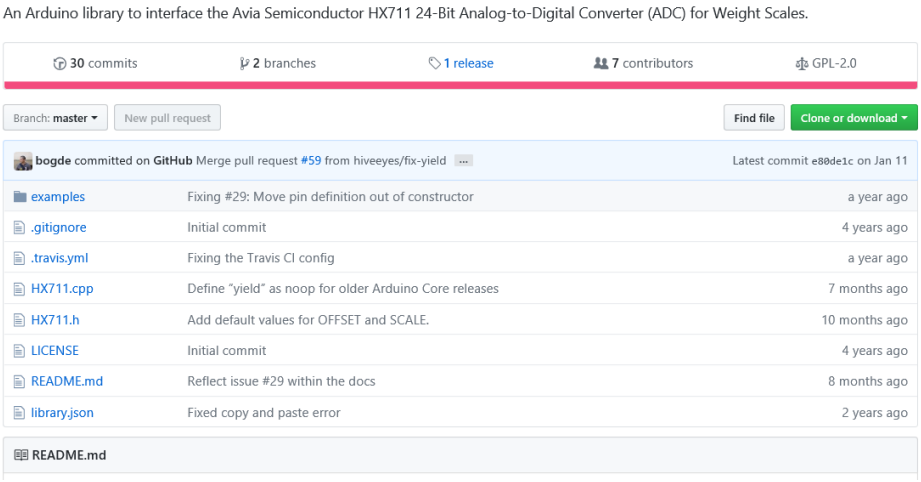
في هذا المثال سيتم استخدام مكتبة HX711 من موقع توجد به كثير من المكتبات النافعة لهواة الأردوينو ومصممي الدوائر الإلكترونية، وهو موقع www.github.com [٣٤]. ويجب على مستخدم أي من هذه المكتبات الرجوع لشرح ومعرفة طريقة عمل هذه المكتبات، حتى تتم الاستفادة القصوى منها. ويمكن إضافة أي من المكتبات إلى برنامج من خلال برنامج بيئة الأردوينو Arduino Environment IDE بالذهاب إلى sketch، ثم include library، ثم manage libraries، أو Add .ZIP library لإضافة أية مكتبة متوفرة لمستخدمي أردوينو [٣٤]. يوضح الشكل (د٣٠) طريقة إضافة مكتبة لأحد برامج أردوينو. بعد إضافة المكتبة يظهر في بداية البرنامج سطر جديد يبدأ بعلامة المربع (#) متبوعًا بكلمة include، ثم بين قوسين زاوية باسم المكتبة. فعلى سبيل المثال، إذا ما تمت أضافة مكتبة HX711 إلى البرنامج سيظهر # include <HX711> في بداية البرنامج، وهذا يعني إضافة المكتبة إلى البرنامج، وعليه فإن المستخدم يستطيع استخدام هذه المكتبة وأوامرها. ولمعرفة أوامر المكتبة وكيفية استخدامها، وجب على المستخدم الرجوع إلى تفاصيل المكتبة، والأمثلة التي توضح كيفية استخدامها من الموقع المشهور [٣٤].



الشكل (د٣٠) إضافة مكتبة لأحد برامج الأردوينو.

يبين الشكل (د٣١) موقع www.github.com [٣٤] ومكتبة HX711 التي يراد استخدامها في هذا المشروع، حيث يمكن تنزيلها على جهاز الكمبيوتر. وحسب مصمم هذه المكتبة، فإن على

المستخدم عمل معايرة باستخدام وزن معروف، ومن ثم إدخال معامل المعايرة في برنامج الأردوينو حتى يتسنى قراءة الوزن صحيحًا.



الشكل (د٣١) موقع github.com وتفاصيل عن مكتبة HX711 [٣٤].

خطوات المعايرة، كما يقترحها مصمم مكتبة HX711، كما يلي:

١- رفع الأوزان من على الميزان وإصدار الأمرين scale.set_scale() و scale.tare() بدون أية قيم، كما يظهر في محتوى البرنامج الأول الموضح في الشكل (د٣٢) الذي يستخدم للمعايرة.

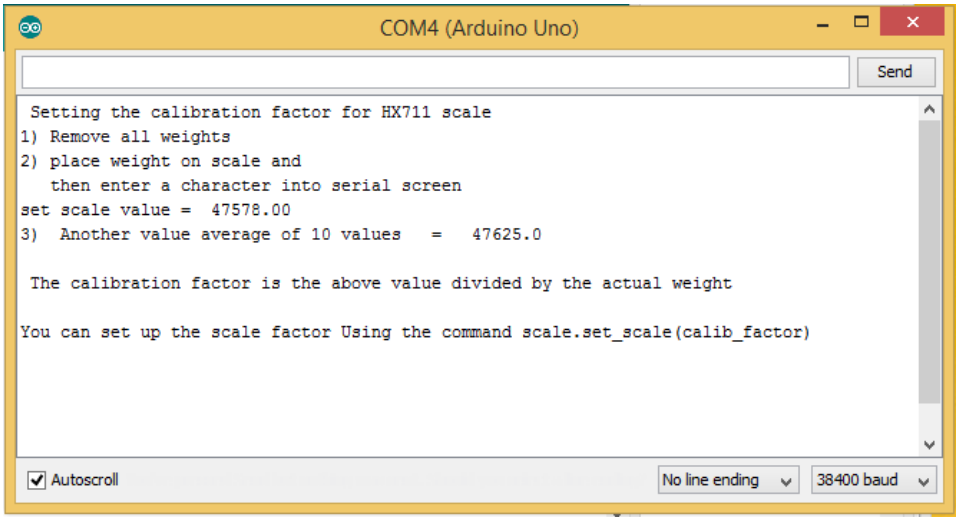
٢- وضع وزن معروف بأفضل دقة ممكنة مثل ١٠٠ جرام على الميزان وقراءة الوزن باستخدام الأمر scale.get_units() وتسجيلها في ورقة خارجية.

٣- قسمة الرقم الذي تم تسجيله في الخطوة رقم ٢ على الوزن المعروف وإيجاد معامل المعايرة calib_f.

٤- إصدار الأمر scale.set_scale(calib_f) حيث معامل المعايرة calib_f هو ما تم الحصول عليه من برنامج المعايرة (انظر الشكل د٣٢) الذي يمثل برنامج الأردوينو الذي تمت كتابته لإيجاد معامل المعايرة. بينما يبين الشكل (د٣٣) شاشة مخرج على التوازي، موضحا عليها قراءة الأوزان باستخدام لوحة الأردوينو ومضخم HX711، وخلية الحمل، وفي حالتنا تم استخدام وزن معروف، وهو ١٠٠ جرام، ومنه بعد تسجل القراءة واتباع الفقرة ٣ أعلاه تم الحصول على معامل المعايرة، وهو في هذه الحالة ٤٧٥,٧٨.

```
#include <HX711.h>          اضافة المكتبة الخاصة بـHX711
// This program is just to find the clibration factor for hx711

/* Steps for perfroming the calibration of hx711
1) In this program you set scale.set_scale() and scale.tare() with no values
2) the program will ask to put a known scale (say 100 g) and a character in serial input
3) you will get a number after you put the scale from scale.get_value (say 50000)
4) divide the value you get in serial port with the known weigth ( i.e. 50000/100=500)
 and insert it into scale.set_scale.
i.e. scale.set_scale(500)
*/
// This program is only to get the calibration factor for HX711

HX711 scale (A1,A0);        تحديد مخرجين على الاردينو لمنظم الساعة و نقل المعلومات

void setup() {
Serial.begin(38400);
Serial.println(" Setting the calibration factor for HX711 scale");
delay(3000);
Serial.println("1) Remove all weights");
delay(5000);
Serial.println("2) place weight on scale and ");
Serial.println("   then enter a character into serial screen");
scale.set_scale();    ترك معامل الوزن بدون قيمة
scale.tare();         ترك معامل التصفير بدون قيمة
delay(7000);
while (!Serial.available()) {};
float x=scale.get_units(5);      متوسط الوزن لخمس قراءات
Serial.print("set scale value =  ");
Serial.print (x);

Serial.println("");
Serial.print("3)  Another value average of 10 values   =   ");
Serial.println(scale.get_units(10),1);  متوسط الوزن لعشر قراءات
Serial.println("");
Serial.println(" The calibration factor is the above value divided by the actual weight");
Serial.println("");
Serial.println("You can set up the scale factor Using the command scale.set_scale(calib_factor)");
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

الشكل (د٣٢) البرنامج المبدئي لإيجاد معامل المعايرة عند استخدام وزن معلوم.



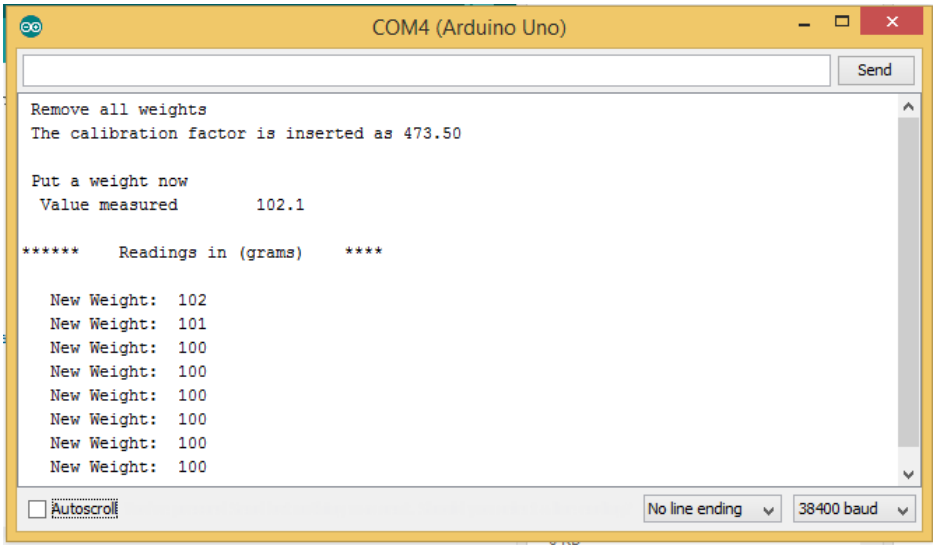
الشكل (د-٣٣) مخرجات برنامج المعايرة لـ HX711.

وبعد المعايرة واستنتاج معامل المعايرة، يمكن إعادة المعايرة مرات أخرى ومحاولة إيجاد معامل المعايرة بدقة أكثر. وفي حالتنا هذه، تمت إعادة برنامج المعايرة مرات كثيرة، وتم الحصول على معامل معايرة (calib_f) يساوي ٤٧٣,٥ وهو فعلا ما تم استخدامه لاحقا.

تمت كتابة برنامج للأردوينو وإدخال هذا المعامل، ومن ثم قياس أي وزن طالما هو في مقدرة خلية الحمل المصمم عليها. ويبين الشكل (د٣٤) برنامج الأردوينو بعد إضافة معامل المعايرة وإضافة جزء متكرر تحت void loop () { }، ويظهر الشكل (د٣٥) نتائج البرنامج عند قياس وزن ١٠٠ جرام.

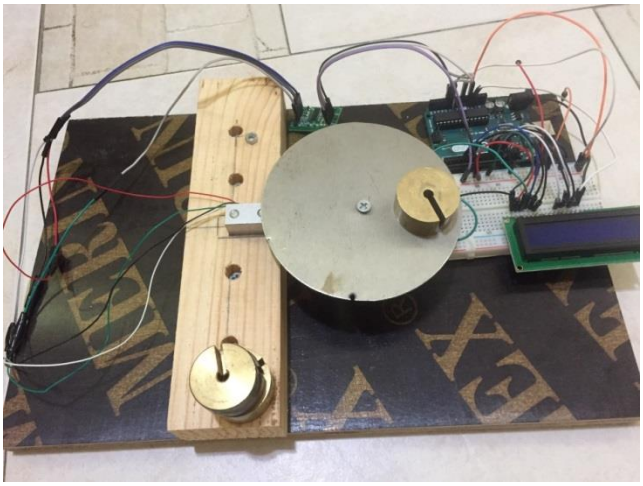
```
#include <HX711.h>
float calib_f=473.5; // This factor was found from a seprate program using a known weight
HX711 scale (A1,A0);

void setup() {
  // put your setup code here, to run once:
Serial.begin(38400);
Serial.println(" Remove all weights");      ارفع جميع الاوزان عن الجهاز
delay(5000);
scale.set_scale(calib_f); هنا تم ادخال معامل المعايرة الذي تم ايجاده سابقا
scale.tare();
Serial.print(" The calibration factor is inserted as ");
Serial.println(calib_f,2);
delay(5000);
Serial.println("");
Serial.println(" Put a weight now"); ضع وزن على الجهاز لمعرفة وزنه
delay(7000);
Serial.print("  Value measured  \t");
Serial.println(scale.get_units(5),1);
Serial.println("");
Serial.println("******    Readings in (grams)    ****");
Serial.println("");
}
void loop() {                                   في هذا الجزء يتم قياس الوزن
  Serial.print("   New Weight:  ");              واظهاره على شاشة التسلسل
  Serial.println(scale.get_units(5),0);
  scale.power_down();
  delay(5000);
  scale.power_up();
}
```

الشكل (د٣٤) برنامج لمعرفة الأوزان بعد المعايرة.

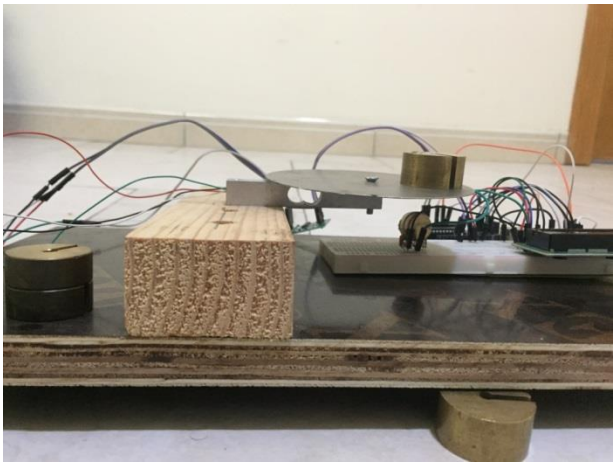


شكل (د٣٥) مخرجات برنامج الوزن بعد المعايرة لوزن ١٠٠ جرام.

يظهر الشكلان (د٣٦ و د٣٧) الميزان الذي تم تصميمه وتنفيذه مع خلية حمل محددة بمقدرة قياس خمسة كيلوجرامات فقط.



شكل (د٣٦) صورة من أعلى للميزان الذي تم تصميمه وتنفيذه باستخدام لوحة أردوينو وخلية الحمل (٥ كجم) والمضخم HX711.



شكل (د٣٧) صورة جانبية للميزان الذي تم تصميمه وتنفيذه باستخدام لوحة أردوينو وخلية الحمل (٥ كجم) والمضخم HX711.

المراجع

[١] Iovine, J., PIC Microcontroller Project Book: For PIC Basic and PIC Basic Pro Compliers, McGraw Hill, 2000.

[٢] Di Jasio, L., Wilmshurst, T., Ibrahim, D., Morton, J., Bates, M., Smith, J., Smith, D. and Hellebuyck, C., PIC Microcontrollers: know it all, Elseiver, 2007.

[٣] Qureshi, M. Amer Iqbal, Teach your self PIC microcontrollers for absolute beginners, ftp://doc.nit.ac.ir/cee/sh.gol/New%20MCU%20LectureNotes/9-Teach Yourself PIC Microcontroller Programming.pdf.

[٤] Milan Verle PIC Microcontrollers- Programming in Basic, MikroElektronika, 2010.

[٥] https://www.parallax.com/catalog/microcontrollers/basic-stamp

[٦] بسيوني، عبد الحميد، "المتحكمات الدقيقة"، دار الكتب العلمية للنشر والتوزيع، القاهرة، مصر، ٢٠٠٨م.

[٧] عبد المعطي، محمد، "بنية وبرمجة المتحكمات المصغرة"، الطبعة الأولى، شعاع للنشر والعلوم، سوريا، ٢٠٠٦م.

[٨] Alciatore, D. G. and Histand, Michael B., Introduction to Mechatronics and Measurement systems, 4th Edition, McGraw Hill, 2012.

[٩] Bishop, R. H., The Mechatronics Handbook, CRC press (www.crcpress.com), 2002.

[١٠] http://www.microchip.com/

[١١] http://www.atmel.com/products/microcontrollers/avr/

[١٢] http://www.ti.com/

[١٣] https://www.renesas.com/en-eu/

[١٤] http://www.st.com/en/microcontrollers.html

[١٥] http://www.intel.com/content/www/us/en/embedded/products/quark/overview.html

[١٦] http://toshiba.semicon-storage.com/us/product/microcomputer.html

[١٧] http://www.nec.com/en/global/solutions/index.html

[١٨] http://www.alldatasheet.com/

[١٩] http://www.datasheetcatalog.com/

[٢٠] https://www.labcenter.com/

[٢١] https://www.mikroe.com/pic/development-boards/

[٢٢] https://www.parallax.com/catalog/microcontrollers/basic-stamp

[٢٣] https://en.wikipedia.org/wiki/Arduino (visited July 29, 2017)

[٢٤] http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino (visited July 29, 2017)

[٢٥] https://www.arduino.cc/

[٢٦] عبد الله، علي عبد الله، الأردوينو ببساطة متوفر مجانا على النت http://simplyarduino.com

[٢٧] الرميض، أسعد، واللاني، علي، والرحيلي، فهد، والمعيري، وائل، احترف الأردوينو في ١٠ أيام، متوفر مجانا على النت.http://librebooks.org/become-an-arduino-professional-in-10-days

[٢٨] Branzi, Massimo, Getting started with Arduino, O'Reilly publishing, 2nd edition, 2011.

[٢٩] https://www.sunfounder.com/learn

[٣٠] https://learn.sparkfun.com/tutorials/what-is-an-arduino

[٣١] https://www.element14.com/community/community/arduino/arduino-tutorials/blog/2014/09/ 04/jeremy-blum-arduino-tutorials

[٣٢] https://www.intorobotics.com/pick-best-temperature-sensor-arduino-project/

[٣٣] https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

[٣٤] https://github.com/bogde/HX711

[٣٥] http://fritzing.org/download/