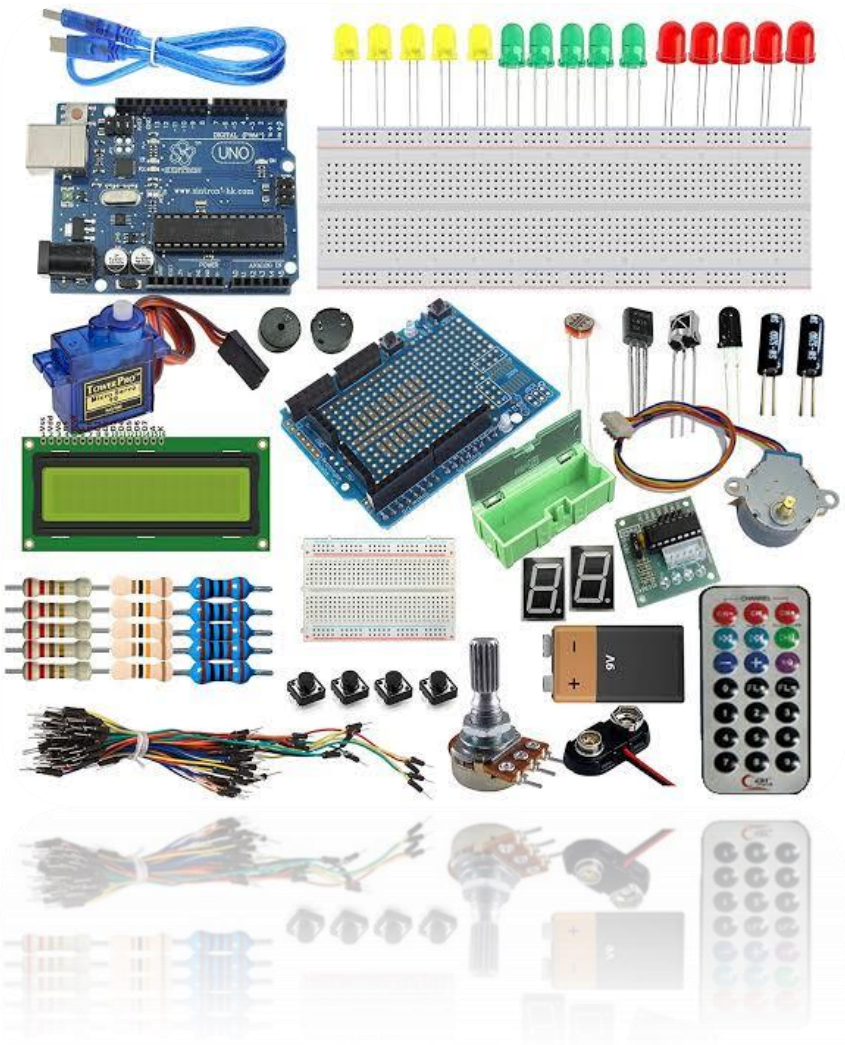
جمهورية مصر العربية
Arab Republic of Egypt

دليــل الطالــب



عنوان الوحدة

الأردوينـــــــــــــــــو

اعداد:

أ. كاملة زايد محمود **أ. عمرو السيد السقعان**

موجه عام مركزي **معلم أول أ**

ملخص الوحدة

تهدف هذه الوحدة الى إكساب الطلاب الجدارات المرتبطة بالتعامل مع الاردوينو من حيث البرمجة ومكوناته الأخرى من حساسات وعناصر الالكترونية لاستخدامه في تنفيذ منظومات تحكم مختلفة .

المادة التعليمية الخاصة بالطالب

مخرجات التعلم

1. ينفذ منظومة تحكم تعمل بالأردوينو باستخدام برامج المحاكاة .
2. يبرمج لوحات الأردوينو.
3. ينفذ منظومات التحكم " **hard wear** " .
4. يراجع أداءه الخاص ويخطط لتحسينه .

مقدمة عن الأردوينو :

أردوينو بالإنجليزية Arduino : هي عبارة عن لوحه إلكترونية ، تتكون من دائرة إلكترونية مفتوحه المصدر مع متحكم دقيق على لوحة واحدة يتم برمجتها عن طريق الكمبيوتر وهي مصممة لجعل عملية استخدام الإلكترونيات التفاعلية في مشاريع متعددة التخصصات أكثر سهولة، ويستخدم الأردوينو بصوره أساسيه في تصميم المشاريع الإلكترونية التفاعلية أو المشاريع التي تستهدف استخدام حساسات مختلفة (مثل درجات الحرارة، الضوء، الضغط..الخ) ويمكن توصيل الأردوينو ببرامج مختلفة علي الحاسب الشخصي ، ويعتمد لوحة الأردوينو في برمجتها علي لغة البرمجة مفتوحه المصدر ، وتتميز الأكواد البرمجية الخاصة بلغه الأردوينو أنها تشبهه لغه سي++ وتعتبر من أسهل لغات البرمجة المستخدمة في كتابه برامج المتحكمات الدقيقة .

شكل (1) يوضح لوحة الاردوينو

و تم تصميم الأردوينو بجزئية : العتاد (hardware) والبرمجي (software) ليقوم بإنشاء مشاريع تفاعلية ويستطيع الأردوينو التفاعل مع الأزرار، الديودات الضوئية، المحركات، مكبرات الصوت، وحدات تحديد المواقع بالأقمار الصناعية (GPS)، الكاميرات، الإنترنت، وحتى الهاتف الذكي .

هذه المرونة في التفاعل بالإضافة إلى أن برنامج الأردوينو مجاني وألواح العتاد الخاصة به رخيصة، وأيضاً سهولة تعلم التعامل مع عتاد الأردوينو وبرمجياته، كل هذا أدى إلى مجتمع ضخم من مستخدمي الأردوينو قاموا بكتابة الأكواد البرمجية والتعليمات الخاصة بعدد ضخم ومتنوع من المشاريع المبنية على الأردوينو لصناعة أي شيء إلكتروني مثل الروبوتات أو أي شيء آخر يمكن استخدام الأردوينو كعقل مفكر في أي مشروع الكتروني.

لذا كان من المهم على فني الأجهزة الإلكترونية أن يكون ملما بهذا المجال الذي أصبح جزءا رئيسيا في صناعة الأجهزة الالكترونية .

المخرج التعليمي (1)	ينفذ منظومة تحكم تعمل بالأردوينو باستخدام برامج المحاكاة .		
معيـــــار الأداء (1)	يثبت التجهيزات البرمجية على جهاز الحاسب الآلي وفقا لمتطلبات التشغيل		
زمن التنفيـــــــــذ		أدلة التعليم	مرفق 1
معلومات إثرائيـــــه	فيديو يوضح التعامل مع القوائم الأساسية في برنامج البروتس Proteus		

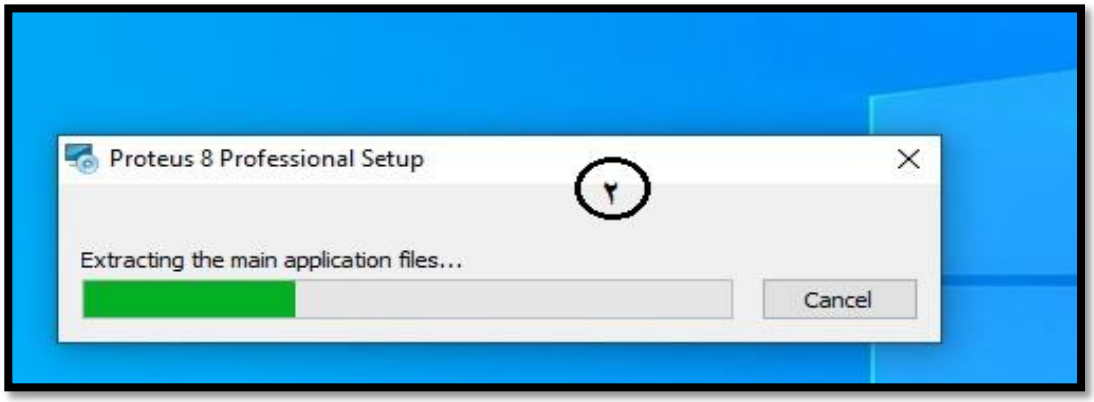
1- برنامج بروتس Proteus :

هو برنامج محاكاة الدوائر الالكترونية ، حيث يوفر بيئة متكاملة تحتوي على كل ما يلزم المستخدم من أدوات لعملية محاكاة واقعية ، فهو يجمع ما بين محاكاة الدوائر والعناصر الإلكترونية ونماذج من بعض المعالجات لتسهيل مرحلة لاحقة من المحاكاة للنظم الإلكترونية المعتمدة على المتحكمات ... وقد كان أول أداة طورت طرق الفحص والمحاكاة لهذه النظم كمرحلة ما قبل التطبيق العملي لمخططات داراتها.

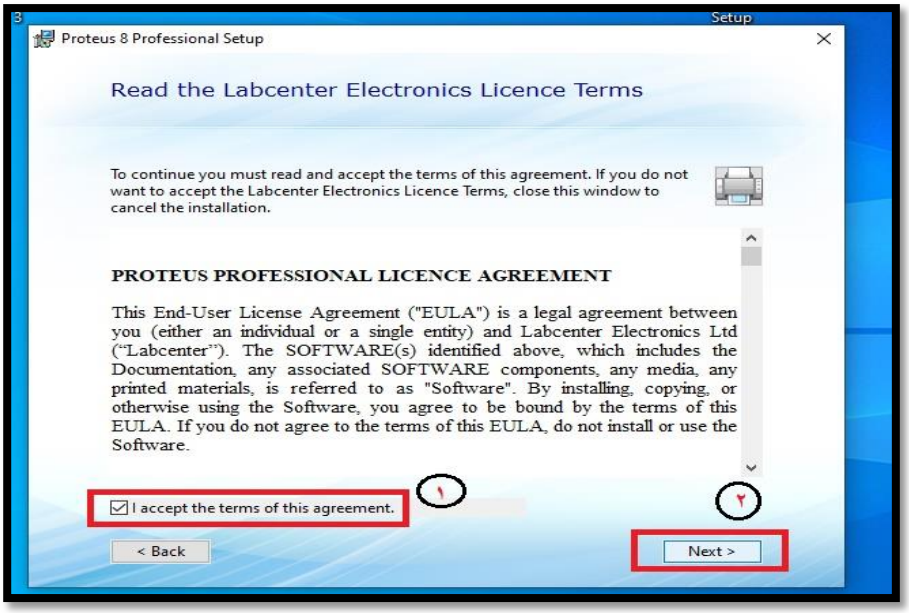
1-1 خطوات تنصيب برنامج Proteus :

- اتبع الخطوات التالية :

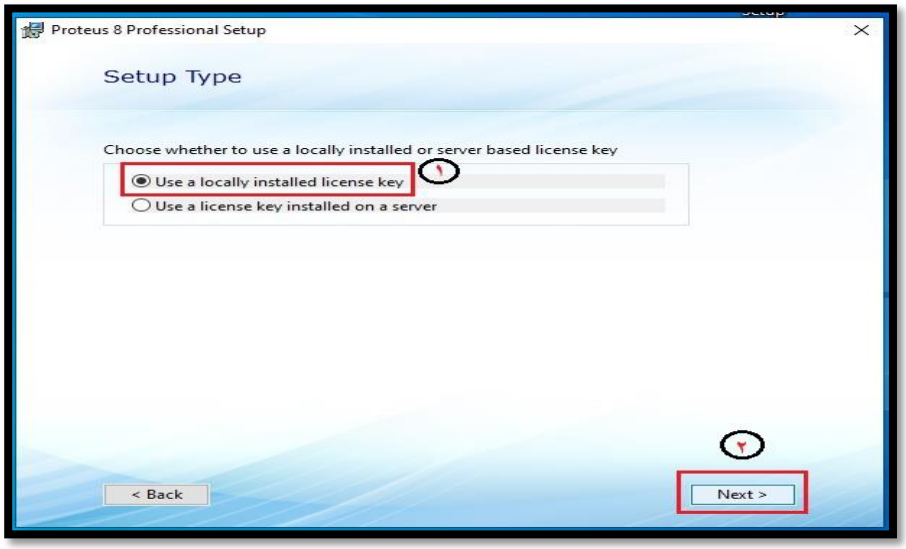
1- يتم الضغط بالزر الايسر للفأرة واختيار Run as administrator



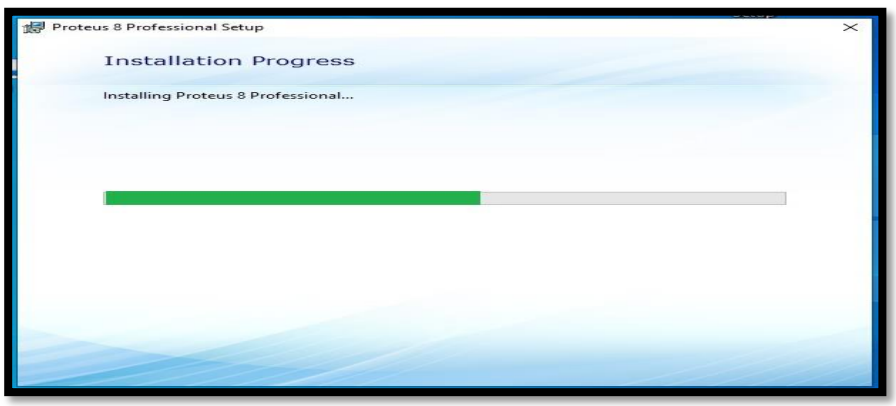
2- يتم استخراج (فك) الملفات الأساسية المضغوطة .



3- ضع علامة صح على قبول الاتفاقية ثم اختر Next



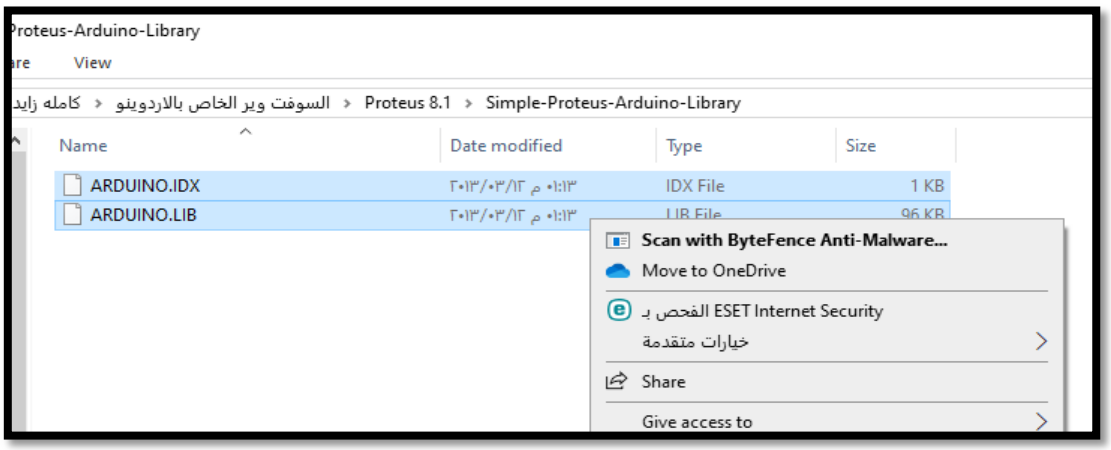
4- ضع علامة على اختيار استخدام المفتاح المرفق مع أسطوانة البرنامج ثم اختر Next



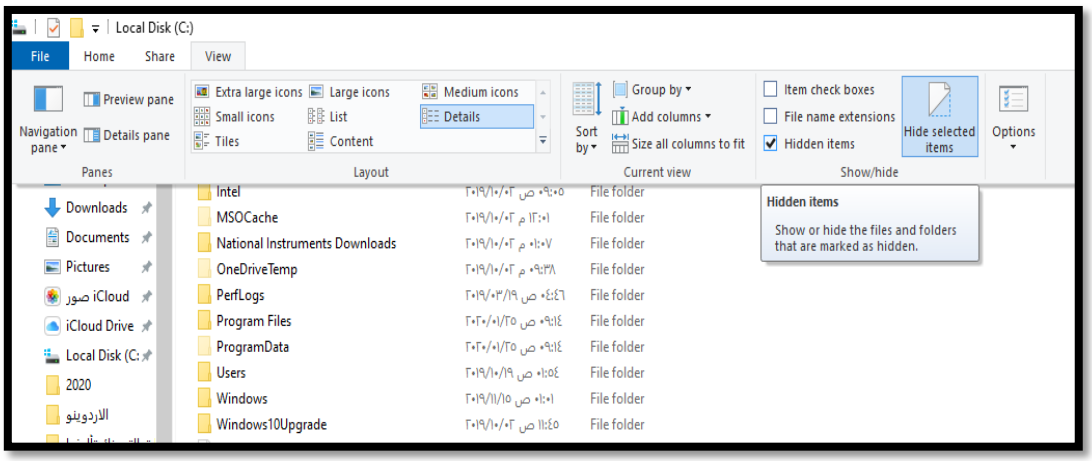
5- واصل عملية التنصيب حتى تظهر أيقونة البرنامج على سطح الكمبيوتر .

2-1 خطوات إضافة مكتبات الاردوينو إلى برنامج Proteus :

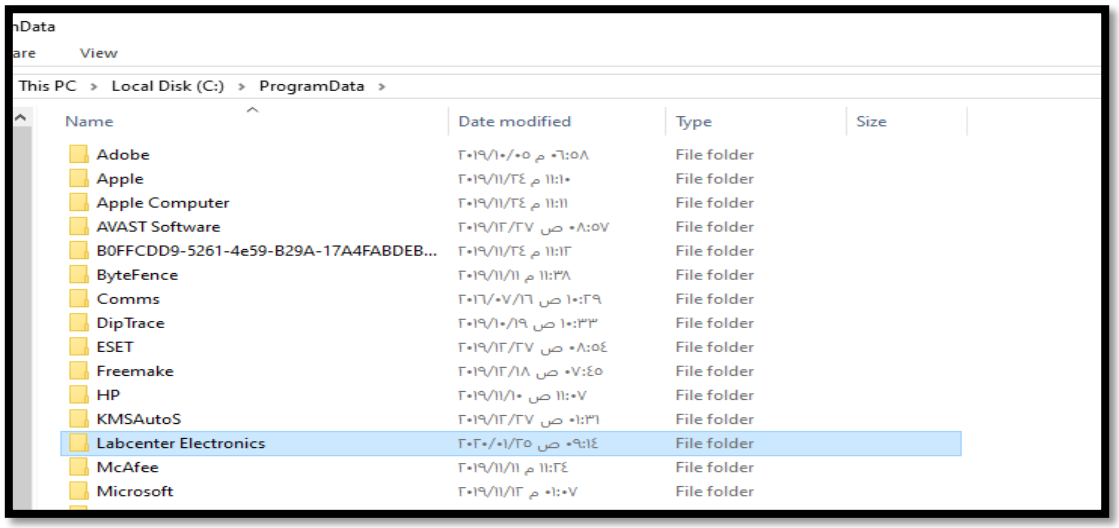
- اتبع الخطوات التالية :



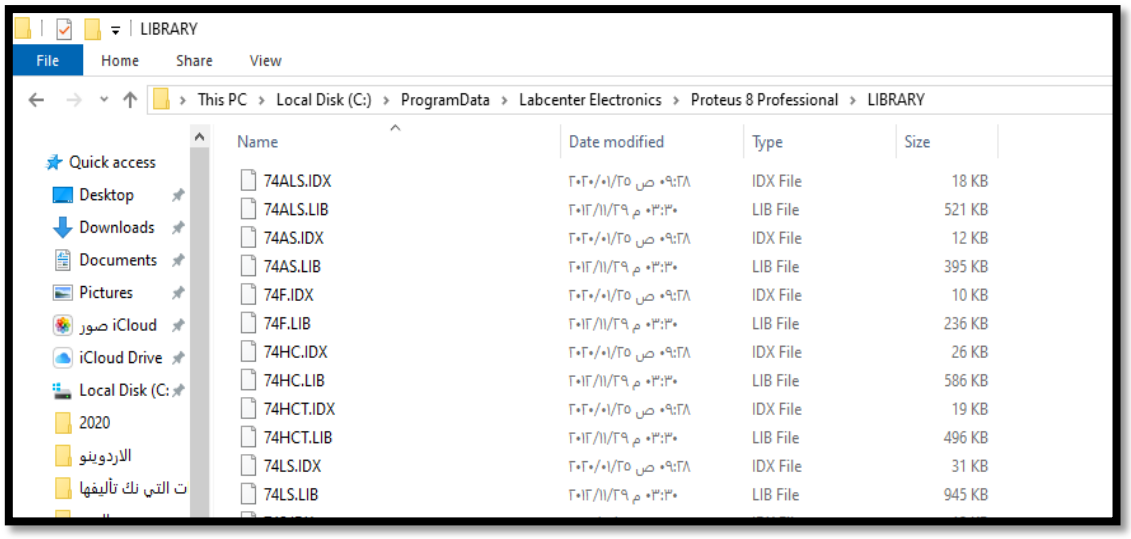
1- قم بنسخ الملفين



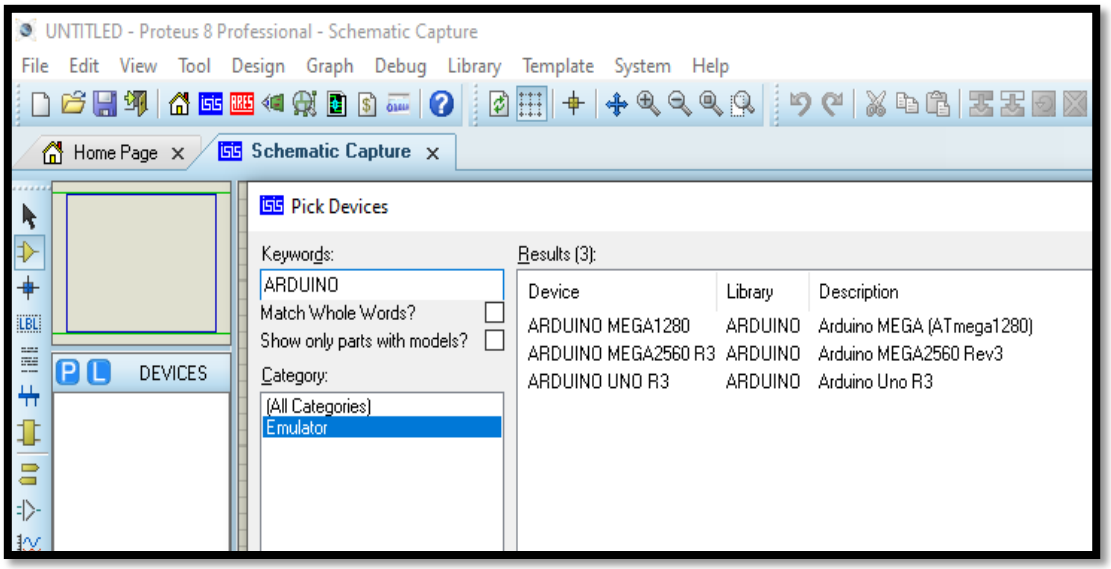
2- أذهب للمحث C - ثم اذهب للقائمة view وأظهر الملفات المخفية



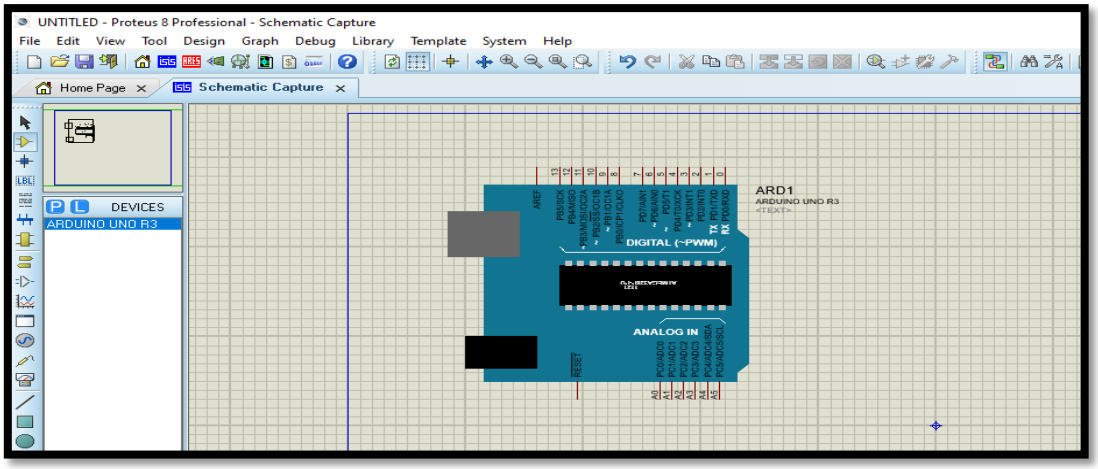
3- اذهب ProgramData ثم افتح الملف Labcenter Electronics



4- ثم Proteus 8 Professional ثم LIBRARY وألصق الملفين

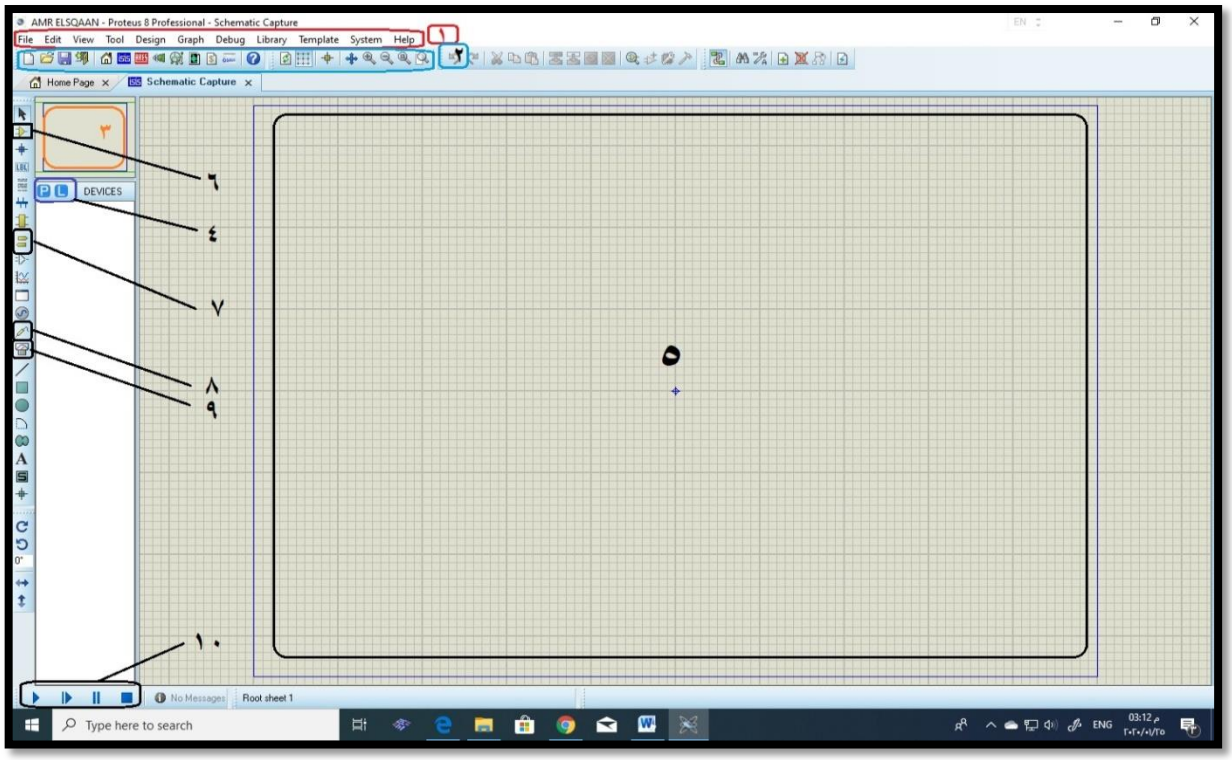


5- افتح البرنامج بروتس واذهب لمكتبة العناصر واكتب ARDUINO



6- اختار بوردة ARDUINO UNO فتظهر في نافذه التصميم

3-1 التعرف على قوائم وواجهات برنامج Proteus :



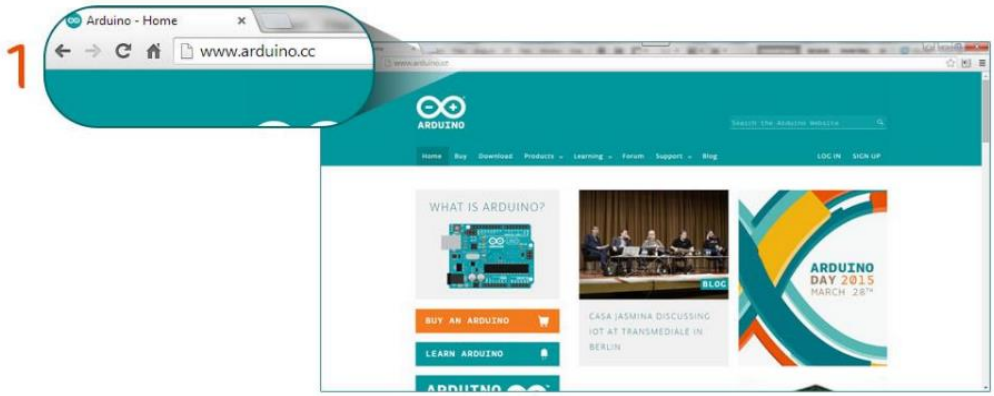
شكل (2) يوضح واجهة Schematic لبرنامج البروتس

-1	شريط القوائم
-2	شريط الأدوات
-3	نافذه المنظر العام للدائرة
-4	أزرار إضافة المكونات الالكترونية
-5	نافذة التحرير : وفيها توضع المكونات وتوصيلها
-6	زر وضع العناصر
-7	أطراف التغذية
-8	مجسات لقياس الجهد والتيار
-9	أجهزه قياس الجهد والتيار
-10	أزرار تشغيل وإيقاف الدائرة

2 تنصيب برنامج الاردوينو IDE والتعرف على واجهاته

يتم برمجة لوحة الأردوينو من خلال برنامج Arduino IED وهو مجاني من شركة أردوينو ويمكن الحصول علي البرنامج وتنزيله من خلال الموقع الرسمي لشركة أردوينو كما يمكن استخدامه مباشرة عبر الموقع الرسمي للشركة :

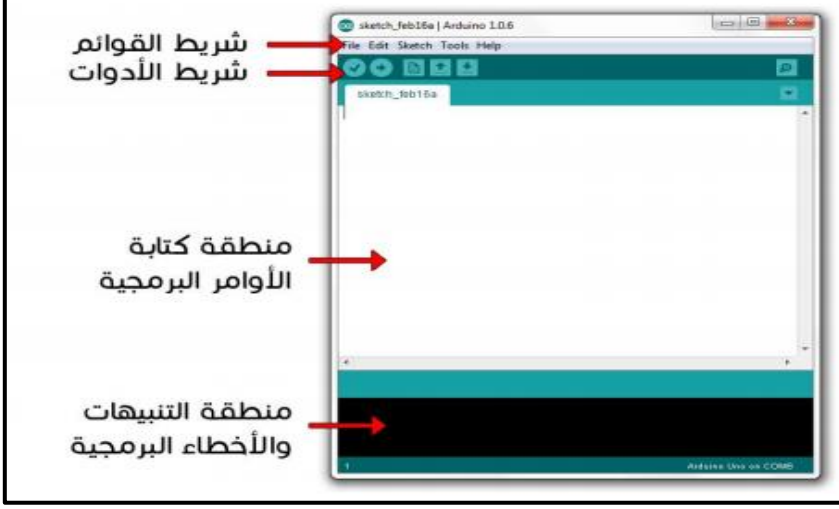
1- اذهب إلى الموقع الرسمي للأردوينو وهو www.arduino.cc .



2- ومن ثم اختر كلمة Download .

3- قم بإختيار نظام التشغيل المناسب لجهازك.

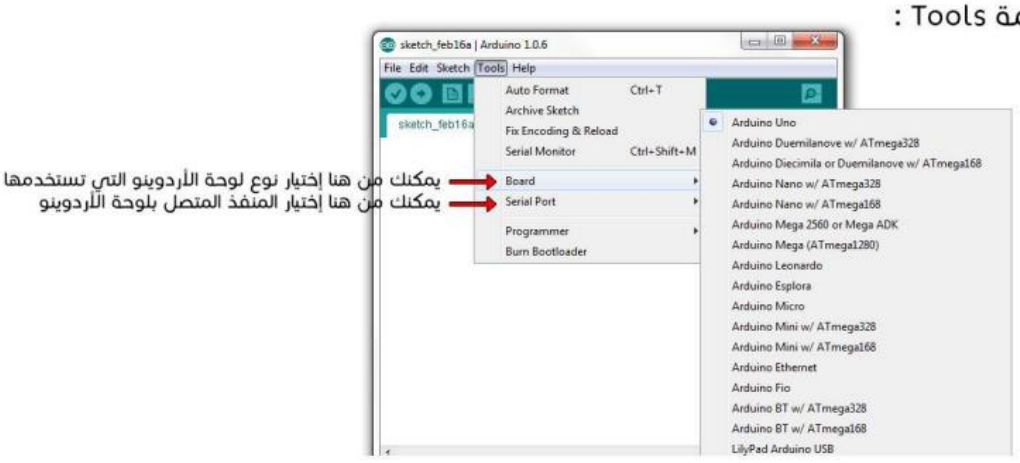
تحتوي واجهة البيئة التطويرية على أربعة أقسام رئيسية كما هو موضح في الصورة:



شريط القوائم :

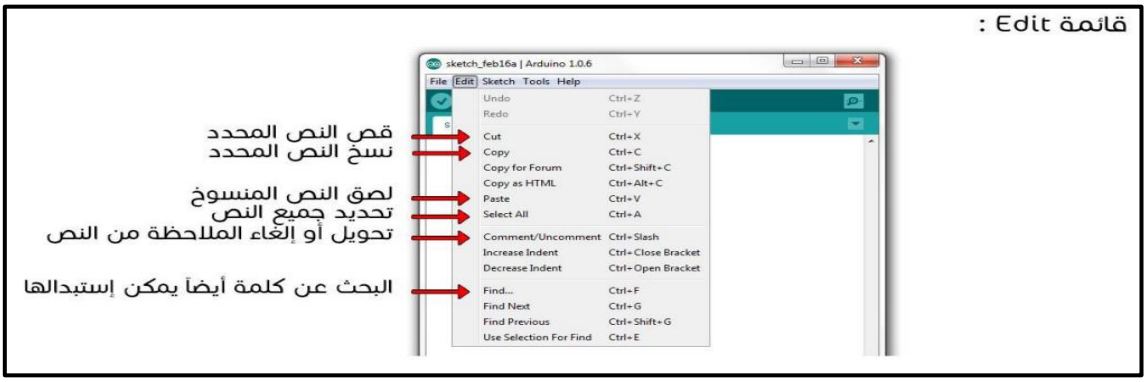
قائمة File :

قائمة Tools :

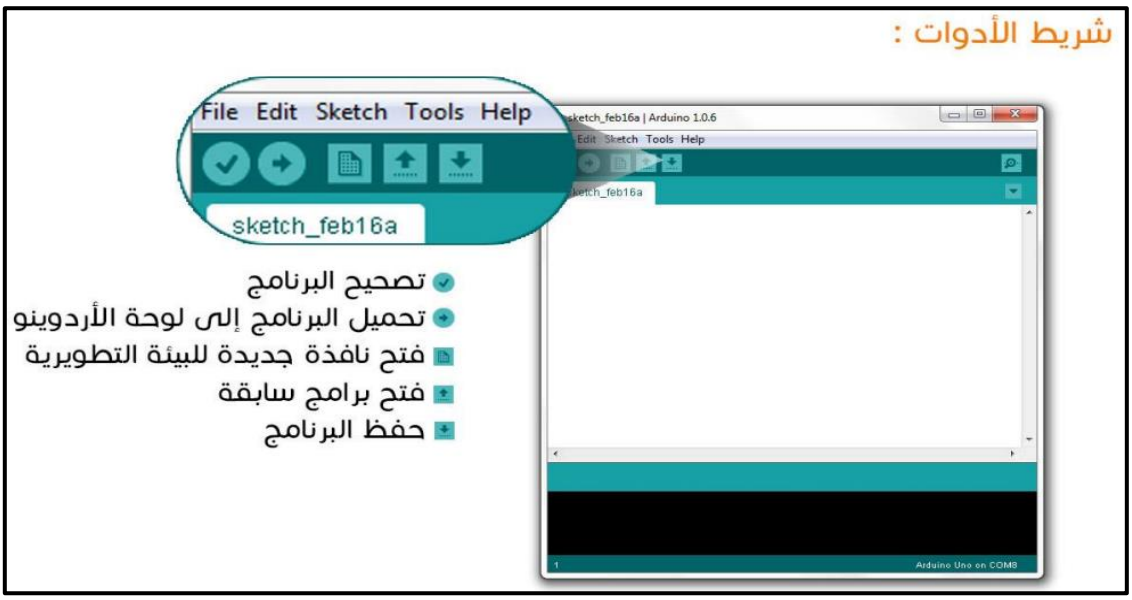


قائمة Sketch :

قائمة Edit :



شريط الأدوات :



مميزات بيئة تطوير أردوينو :

- **البساطة والسهولة في التعامل**
- **خلوها تقريباً من أي تعقيدات في المظهر العام**
- **تحتوي فقط على ما يحتاجه المبرمج ليبدأ تطوير برامج بلغة أردوينو C**
- **تستخدم مباشرة لرفع البرنامج بالصيغة التنفيذية للبوردة (المتحكمة الدقيقة) وبالتالي لن تحتاج لبرنامج آخر**
- **تتوفر بيئة تطوير أردوينو لجميع أنظمة التشغيل المختلفة**

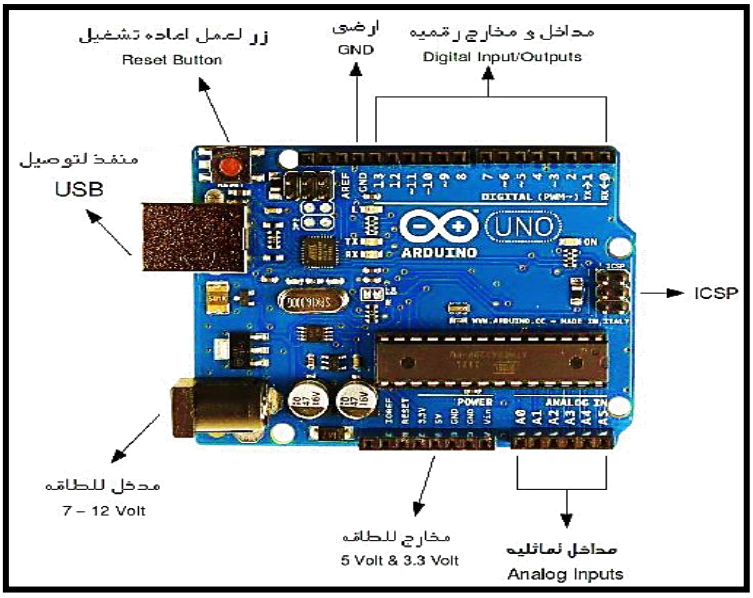
المخرج التعليمي (2)	يبرمج لوحات الأردوينو		
معيـــــار الأداء (1) معيـــــار الأداء (2)	يقارن لوحات الأردوينو بلوحات الأنظمة الأخرى وفقا للمعايير القياسية . يحدد الحساسات والموديلات وفقا لمتطلبات الدائرة .		
زمن التنفيـــــــذ		أدلة التعليم	مرفق 4
معلومات إثرائيـــــه	مقطع فيديو يوضح المقارنة بين لوحة الأردوينو ولوحة الراسبيري باي		

- **أنواع لوحات الاردوينو :**

هناك العديد من أنواع لوحات الأردوينو ولكلا منهم خصائصه ومواصفاته لذا سوف نكتفي بذكر الاتي

1- أردوينو أونو (Arduino Uno)

- هي دائرة إلكترونية صغيرة تستخدم في برمجة متحكمة من شركة أتمل ATmega328
- توفر هذه الدائرة 14 (مدخل / مخرج) منفذ رقمي لتوصيل المكونات الإلكترونية بالمتحكمة
- يمكن استخدام 6 أطراف من هذه المنافذ كمخارج PWM (التعديل الرقمي المعتمد على عرض النبضة)
- تحتوي على مذبذب كريستال بتردد 16MHz
- تحتوي على مدخل USB للتواصل مع الحاسب
- لها مدخل منفصل للطاقة
- لها أيضاً ICSP header (طريقة إضافية لبرمجة المتحكمة وهي لاتزال موصلة بالبوردة بخلاف USB)
- وتعتبر بوردة اردوينو هذه بوردة تطوير ومبرمجة مصغرة ومُهيئة للاستخدام للمباشر



شكل (3) يوضح لوحة الاردوينو أنو

☒ **طرق امداد لوحة Arduino Uno بالطاقة :**

1- امداد اللوحة بالطاقة من خلال منفذ الـ USB

2- امداد اللوحة بالطاقة عن طريق مصدر تغذية خارجي 9V DC

☒ **مخارج الطاقة (الجهد) للوحة Arduino Uno :**

1- **5V :** جهد منتظم يستخدم لتأمين الطاقة للميكروكنترولر وغيره من العناصر على الدائرة. يأتي هذا الجهد من خلال Vin عبر منظم جهد داخلي أو تأمينه من خلال منفذ الـ USB أو أي مصدر جهد منتظم 5 فولت.

2- **3.3V :** مصدر جهد بقيمة 3.3 فولت مؤمن من قبل منظم الجهد الداخلي للدائرة وأقصى قيمة لسحب التيار من خلال هذا الخط هو 50 ميلي أمبير.

3- **GND :** الخط الأرضي للدائرة.

☒ **مداخل ومخارج التحكم في الاردوينو**

1- يوجد عدد أربعة عشر منفذ رقمي معنونه من D0 الى D13 تستخدم كـ (الدخل/ الخرج) وذلك باستخدام الأوامر البرمجية وتعمل هذه الخطوط على جهد أقصاه 5V وكل خط يمكن أن يؤمن سحب للتيار بحدود 40 ميللي أمبير

2- يوجد 6 منافذ دخل تماثلية ومعنونه من A0 إلى A5 بشكل افتراضي وتستطيع هذه المداخل قياس جهد من 0 إلى 5 فولت.

مميزات بوردات اردوينو:

1- لوحات الاردوينو مُصممه لتُناسَب احتياجات الجميع.

2- لوحة الأردوينو أقل ثمنًا مقارنَة بأي منافس لها و سهولة التعامل وتوصيل الدوائر بها

3- برنامج الأردوينو له القدرة على العمل مع جميع أنظمة التشغيل المختلفة

4- بيئة برمجية سهلة حيث ان لغة "Arduino C"سهلة التعلم.

5- توفر ملحقات عديدة لها

6- توفر مكتبات لأغلب ملحقاته على سرعة تطويره

7- مفتوح المصدر مما يساعد

عيوب بوردات اردوينو:

1- تعتبر القدرة البرمجية للأردوينو أقل بكثير من الكمبيوتر ،

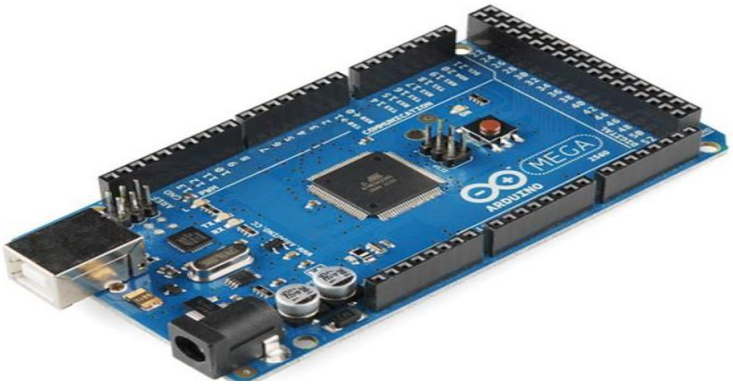
2- لا يتمكن من تشغيل برامج مثل الويندوز أو الأندرويد أو تشغيل شاشات ذات وضوح عالي و هكذا .

3- التيار الذي يتمكن الأردوينو من إخراجه من المنافذ = 20 mA وهذا لا يكفي لتشغيل محرك أو مرحل ريلاي عادة . لذا يجب استخدام عناصر إلكترونية لتكبير الطاقة الكهربائية في بعض التطبيقات

2- أردوينو ميجا (Arduino Mega)

أردوينو ميجا هو الأخ الأكبر لأردوينو أونو. فهو يحتوي على الكثير من منافذ (الدخل/ الخرج) الرقمي (14 منها يمكن استخدامها كمخارج ذات قدرة على تعديل عرض النبضة)، بالإضافة إلى 16 منفذ دخل تناظري، وصلة USB، مقبس للطاقة، وزر لإعادة الضبط.

العدد الكبير من المنافذ الموجود في هذا اللوح يجعله مناسباً جداً للمشاريع التي تتطلب الكثير من منافذ (الخرج/ الدخل) الرقمية (مثل المشاريع التي تحتوي على الكثير من الديودات الضوئية).



شكل (4) يوضح لوحة الاردوينو ميجا

3- أردوينو ليوناردو (Arduino Leonardo)

أردوينو ليوناردو هو أول لوح أردوينو مطور يستخدم متحكم دقيق واحد مع منفذ USB مدمج. أيضاً بسبب تعامل اللوح مع USB بشكل مباشر فإن مكتبات الأكواد (code libraries) متاحة لتسمح للوح بالاتصال بالكمبيوتر كفأرة أو لوحة مفاتيح كما أنة يتميز برخص سعرة وبساطة التركيب .

شكل (5) يوضح لوحة الاردوينو ليوناردو

مقارنة لوحات الأردوينو بلوحات الانظمة الاخرى :

م	وجه المقارنة	لوحة الاردوينو (موديل أونو)	لوحة الراسبيري باي (موديل B)
-1	سرعة المعالج بالميجاهرتز	من 16 إلى 20	من 700 إلى 1000
-2	الذاكرة العشوائية	2KB	512MB
-3	الذاكرة الثابتة	32KB	ذاكرة SD خارجية
-4	جهد تشغيل اللوحة	من 7 إلى 12 فولت	5 فولت
-5	أقل تيار لتشغيل اللوحة	32 mA	من 320 mA إلى 700 mA
-6	عدد مخارج التحكم الرقمي	14	8
-7	المداخل التماثلية	يوجد	لا يوجد
-8	مخارج التعديل النبضي PWM	6	1
-9	توصيل أجهزة إضافية على USB	لا يوجد	يوجد عدد 2 منافذ USB
-10	مدخل توصيل بالشبكة	لا يوجد	يوجد
-11	مخرج الصوت	لا يوجد	يوجد
-12	تشغيل فيديو عالي الدقة	لا يوجد	يوجد

- **أيهما تستخدم لوحة الاردوينو أو لوحة الراسبيري باي ؟**

الاردوينو: إذا كنت تبحث عن السهولة والسعر الرخيص

الاردوينو: إذا كنت تبحث عن استهلاك الطاقة

الراسبيري باي: إذا كنت تبحث عن القوة

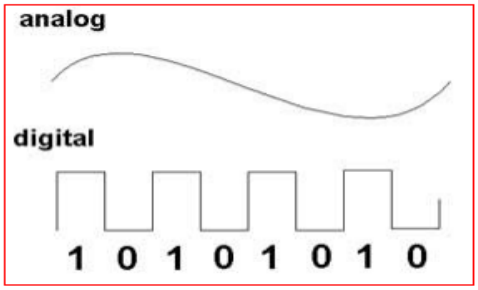
الراسبيري باي: إذا كنت تبحث عن تعدد المهام والعمليات في وقت واحد

الراسبيري باي: إذا كنت تبحث عن توصيل أجهزة ميديا مثل (السماعات – الكاميرات – الشاشات)

- **الفرق بين الاشارات الرقمية والتماثلية :**

-الاشارة التماثلية : هي اشارة كهربية متغيرة أي لها قيم مختلفة ومتصلة بالزمن.

- الاشارة الرقمية هي إشارة لها قيم محددة مسبقا ولا يمكن مجاوزه هذه القيم مطلقا وعادة ما تكون قيمتين فقط (1,0) مثال للتفرقة بين التماثلي و الرقمي : الرقمي مثل اللمبات العادية (OFF-ON) فقط أما التماثلي مثل اللمبات التي تقبل التحكم بشدة الإضاءة .

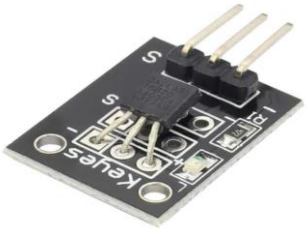
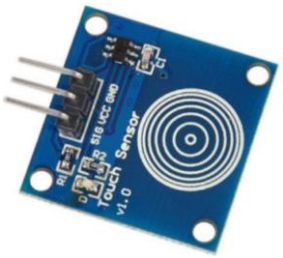
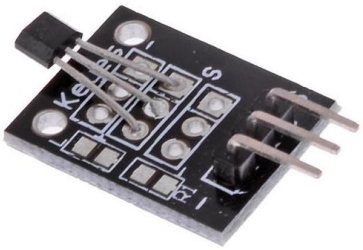


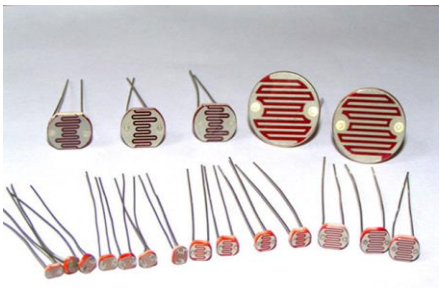
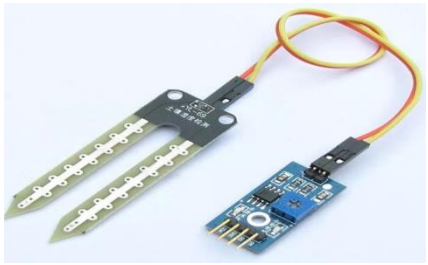
الحساسات والموديولات المستخدمة في التحكم بالاردوينو

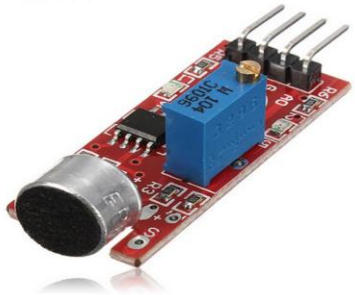
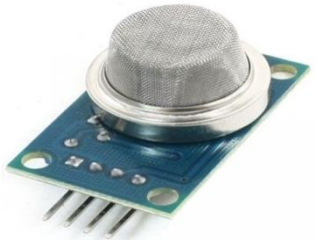
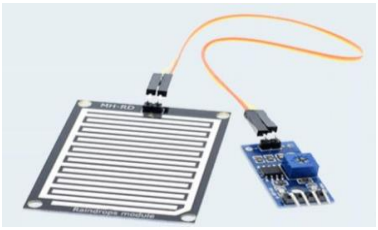
أولا : الحساسات :

الحساس : هو جهاز يحول المقادير الفيزيائية إلى مقادير كهربية من(حرارة – ضغط – إضاءة .....) إلى (جهد – تيار – مقاومة) وتنقسم الحساسات إلى حساسات تناظرية وحساسات رقمية .

وتأتي الحساسات المستخدمة في الأردوينو على شكلين – الأول في صورة منفردة مثل حساس الحرارة lm35 والمقاومة الضوئية LDR والصورة الثانية يأتي داخل لوحة إلكترونية مزود بها دائرة تكبير وتغذية مناسبة لتشغيل الحساس والشكل (6) يوضح مجموعة من الحساسات المستخدمة مع لوحة الاردوينو .

م	اسم الحساس ووصفه	الشكل
1	حساس الحرارة : يستخدم الترانزستور DS18B20 بحيث يوفر قراءة رقميه عن طريق استخدام مخرج واحد فقط	
2	حساس اللمس Touch sensor : حساس اللمس في العمل هي نفس طريقة المفتاح العادي فعندما تقوم بالضغط على حساس اللمس ، فأنت تقوم بأغلاق الدائرة الكهربائية ، سامحاً للتيار بالتدفق وعندما تقوم برفع الضغط عن حساس اللمس ،فأنت تقوم بفتح الدائرة الكهربائية مما يؤدي الى وقف تدفق التيار الكهربائي	
3	حساس تدفق سريان الماء: Water flow sensor : حساس تدفق السوائل هو جهاز يستخدم لقياس معدل التدفق أو كمية التحرك من السائل و التي تمر من خلال أنبوبة أو ماسورة .	
4	حساس المجال المغنطيسي Linear Magnetic Hall Sensor هذا الحساس هو حساس تناظري يعطي إشارة على شكل فولت تتغير بتغير قوة المجال المغناطيسي المحيط و عند توصيله بلوحة الاردوينو يمكن قياس هذا الفولت.	

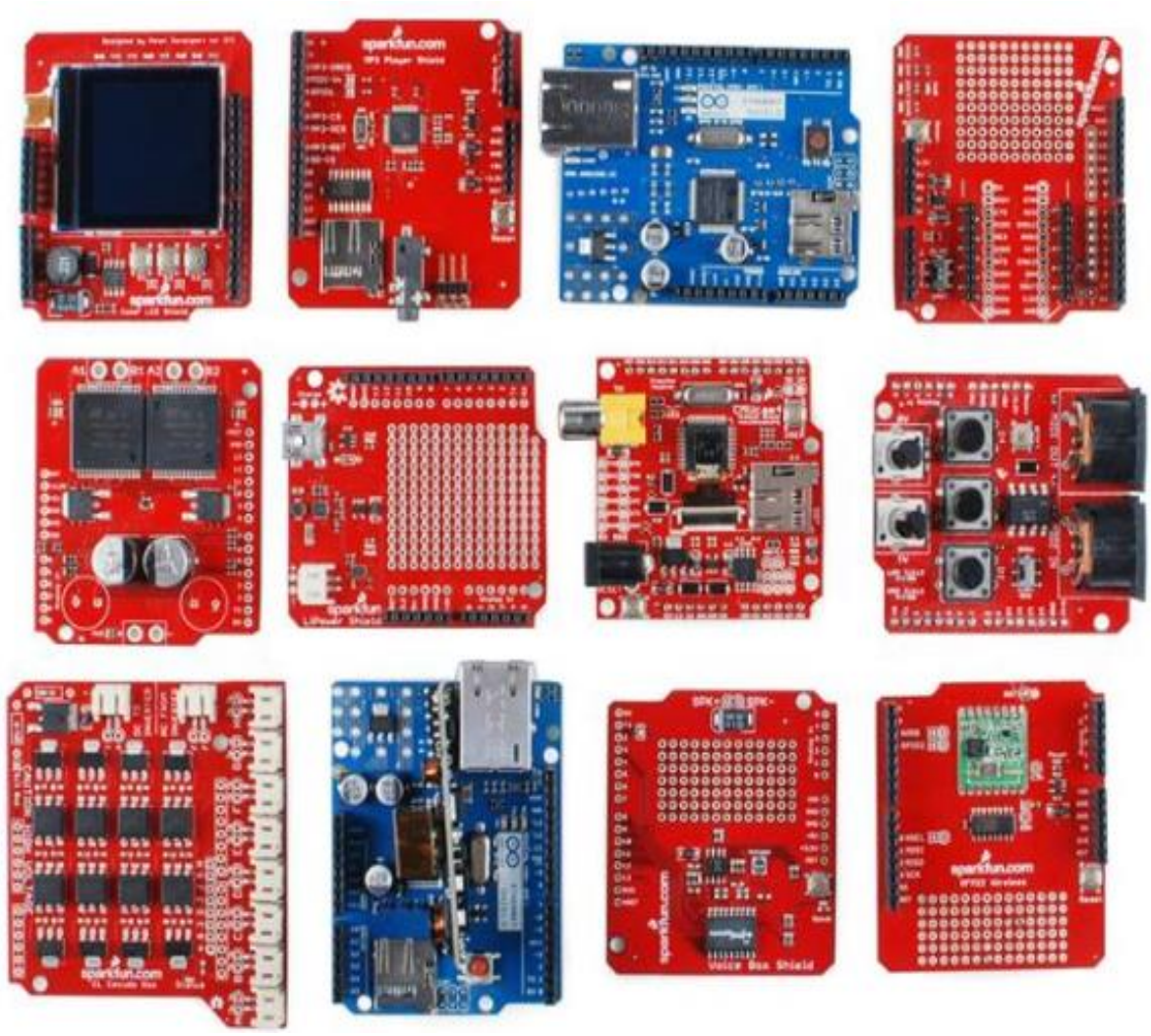
م	اسم الحساس ووصفه	الشكل
5	**حساس الضوء(Light -dependent resistor)** المقاومة الضوئية هي عبارة عن مقاومة تتغير قيمتها اعتمادا على مقدار الضوء المسلط عليها , فكلما زاد مقدار الضوء كلما قلت المقاومة والعكس بالعكس صحيح , تعتبر المقاومة الضوئية مفيدة جدا في كثير من الاجهزة والتطبيقات التي تعتمد علي الضوء.	
6	**حساس الحركة (PIR motion sensors)** حساس الحركة هو في الأساس عبارة عن جهاز قادر على حساب الأشعة تحت الحمراء . فعندما يتم تشغيل الحساس للحركة فانه يحسب مقدار الأشعة تحت الحمراء المحيطة به, وعندما يحدث تغير بمقدار الأشعة تحت الحمراء فإنه يبعث اشارة كهربائية إلى الأردوينو على سبيل المثال , يستخدم حساس الحركة للكشف عن اي حركة تمت في محيط الحساس	
7	**حساس الأشعة تحت الحمراء:(IR sensor)** يستخدم هذا الحساس في التحكم عن بعد بأجهزة معينة وهو يحتوي على باعث للأشعة تحت الحمراء ليقوم بأرسال الاشارات , وبالطبع يمكننا استخدام هذه الاشارات كإشارة إدخال للمتحكم للاردوينو .	
8	**حساس الموجات الصوتية Ultrasonic Sensor HC-SR04** يمكن استخدام هذه الحساسات في التطبيقات الصناعية مثل استخدامها في الأنسان الآلي ليستطيع تحديد المسافة بينه وبين الحواجز من حوله، أيضاً يتم استخدامها في التطبيقات الطبية مثل أجهزة السونار وأجهزة الأشعة الفوق صوتية.	
9	**حساس قياس رطوبة التربة: Soil Moisture** حساس قياس الرطوبة Moisture sensor هو ببساطة يسمح بمعرفة حالة التربة من حيث الرطوبة أو الجفاف حول الحساس , وهو بذلك يستخدم في المساعدة في سقي الاشجار والزهور في الحديقة من دون تتدخل الانسان.	

م	اسم الحساس ووصفه	الشكل
10	**حساس الصوت** الحساسات الصوتية هي التي تعمل على استشعار الصوت والصوت عبارة عن حركة الجزيئات عامة والهواء خاصة. والحساس يستخدم لتعرف علي أي أصوات في الجوار أو بقرب الحساس	
11	**حساس الغازات: MQ2 Gas Sensor** هو عباره عن حساس يقوم بالكشف عن الغازات المحيطة به وهذا الحساس له قدرة عالية على قياس تركيز الغازات القابلة للاشتعال مثل غاز الميثان و البروبان .	
12	**حساس الالوانTCS3200 Color Sensor** حساس الالون هو حساس يستطيع التمييز بين الالوان المختلفة ، و يستطيع ان يحول شده الالوان الى ترددات مختلفة تعبر عن شده و نوعية اللون.	
13	**حساس اللهب: Flame Sensor** حساس أو مستشعر اللهب , فهو يعتمد على التقاط الطول الموجي للضوء و المتولد في الأشعة تحت الحمراء التي يطلقها لهب النار. و يعد هذا الحساس مثالياً في تطبيقات اكتشاف الحرائق و يتميز بوجود مخرج رقمي و أخر تناظري إلى جانب تواجد مقاومة متغيرة لضبط حساسيته.	
	حساس المطر: Rain Sensor يشعر هذا الحساس بكمية المياه المتساقطة عليه وبالتالي يزيد خرج الحساس كلما زادت كمية تدفق المياه علية	

شكل (6) يوضح مجموعة من الحساسات المستخدمة مع لوحة الاردوينو

ثانيا : المديولات :

وهي بشكل أساسي ألواح دوائر إلكترونية مسبقة البناء يتم تركيبها أعلى ألواح الأردوينو لتزويدها بإمكانيات إضافية مثل التحكم بالمحركات، الاتصال بالإنترنت، السماح بالاتصال الهاتفي وإرسال الرسائل عن طريق شبكة المحمول، أو الاتصال اللاسلكي، التحكم في شاشات LCD، وغير ذلك الكثير. والشكل (7) يوضح مجموعة من الموديلات المستخدمة مع لوحة الأردوينو



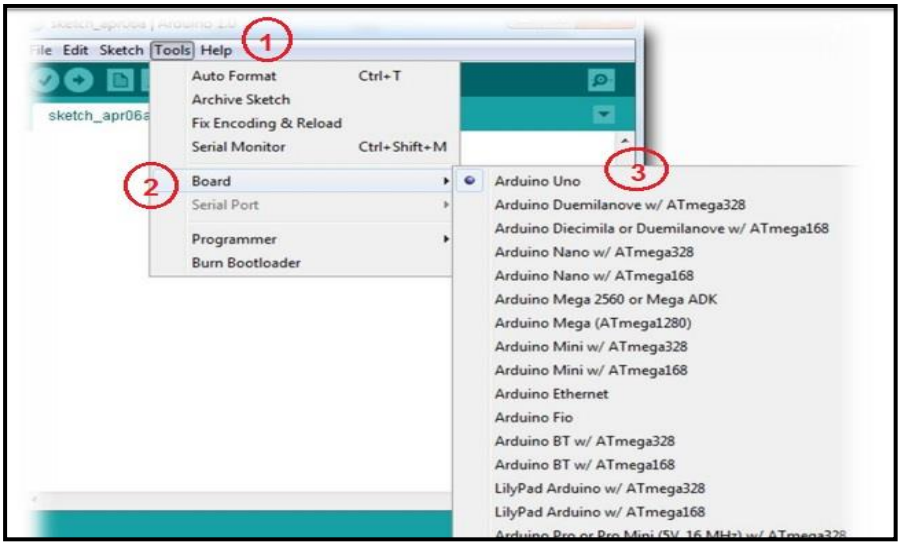
شكل (7) يوضح مجموعة من الموديلات المستخدمة مع لوحة الأردوينو

المخرج التعليمي (2)	يبرمج لوحات الأردوينو.		
معيـــــار الأداء (3)	نقل الملف التنفيذي على لوحة الأردوينو وفقا للتطبيق المطلوب		
زمن التنفيـــــــذ		أدلة التعليم	مرفق 5
معلومات إثرائيـــــه			

بعد كتابة البرنامج (الكود البرمجي) في Arduino IED تأتي مرحلة تحويل هذا الكود من لغة البرمجة C++ أو لغة C Arduino إلى ملف تنفيذي بالنظام السداسي عشر لنقله الى المتحكمة الموجودة بالأردوينو أو المتحكمة الموجودة في برنامج المحاكاة لتنفيذ البرنامج ويتم ذلك من خلال الخطوات التالية :

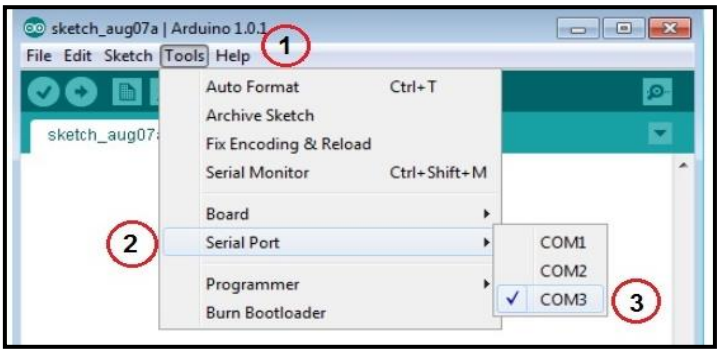
أولا : تجهيز بيئة Arduino IED :

1- في البداية يجب علينا اختيار نوع لوحة الأردوينو المراد تنفيذ المشروع عليها وغالبا ما تستخدم بوردة Arduino Uno والشكل (9) يوضح ذلك .



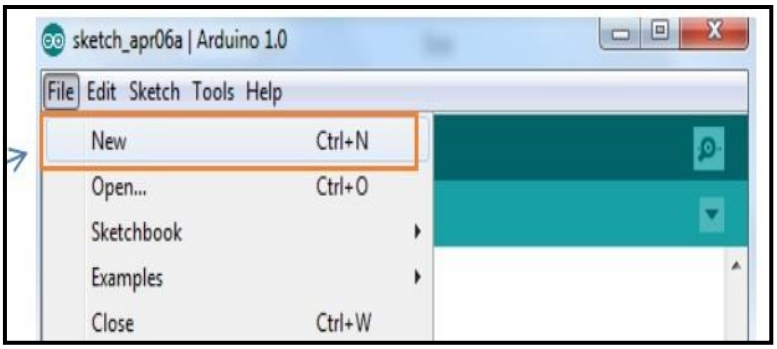
شكل (9) يوضح خطوات اختيار نوع بورده الاردوينو

2- الخطوة التالية هي اختيار منفذ التوصيل بلوحة الأردوينو من قائمة Tools-Serial Port

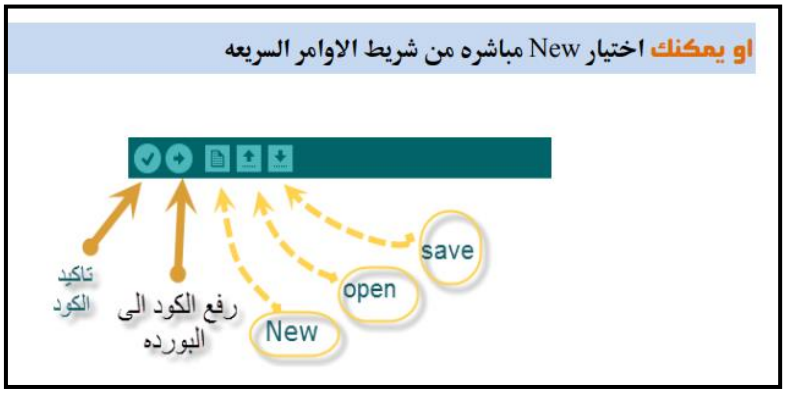


شكل (10) يوضح اختيار منفذ التوصيل بلوحة الاردوينو

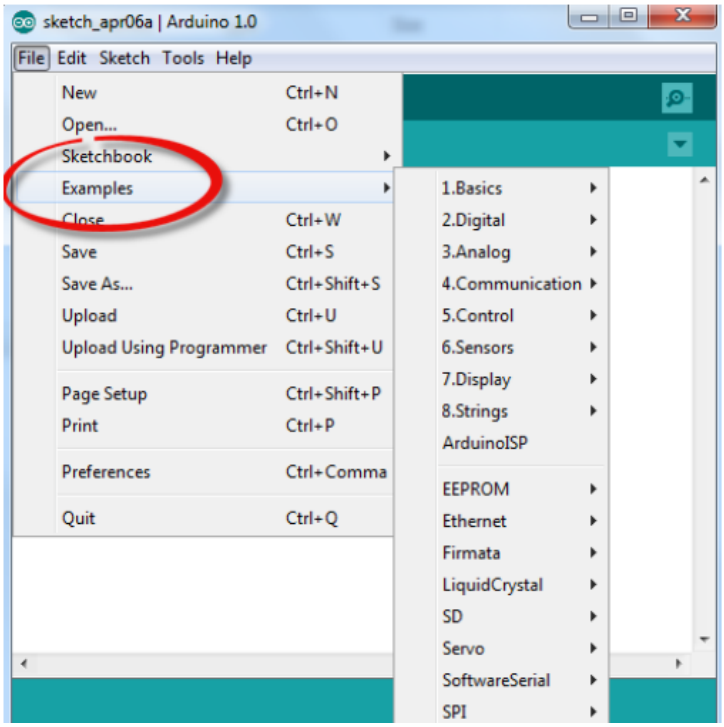
3- نستطيع البدء في كتابة البرنامج وذلك عن طريق اختيار NEW من قائمة File والشكل (11) يوضح ذلك



الشكل (11) يوضح خطوات اختيار صفحة جديدة لكتابة البرنامج



4- يوجد مجموعة من الامثلة البرمجية الجاهزة توفرها البيئة التطويرية للأردوينو والتي تستطيع الوصول اليها من خلال قائمة Examples الموجودة في قائمة File الرئيسية كما في الشكل (12)



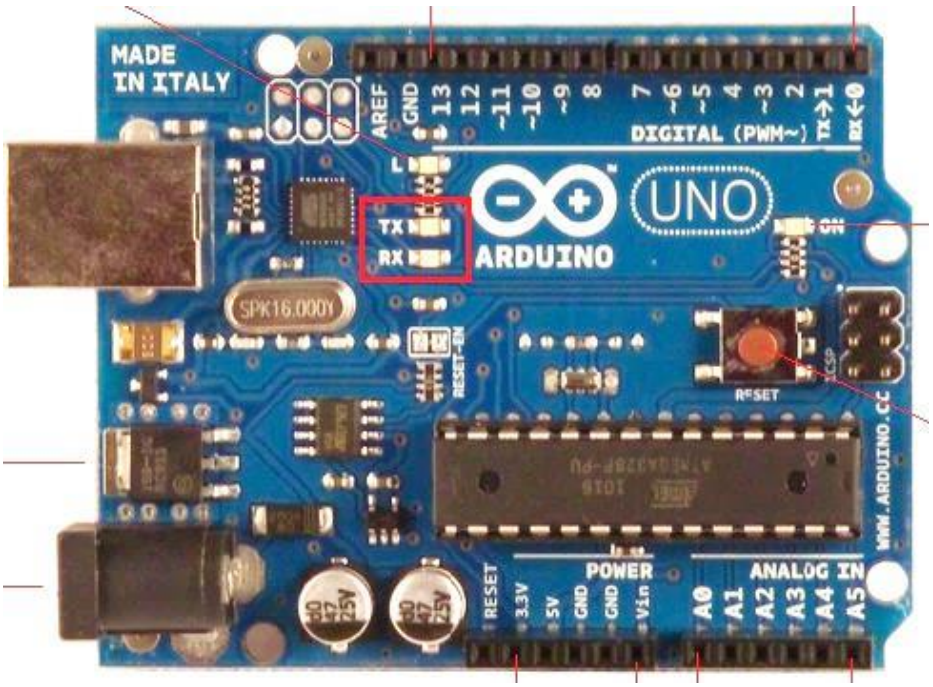
شكل (12) يوضح خطوات اختيار الامثلة البرمجية الجاهزة

ثانيا : نقل الملف التنفيذي على لوحة الأردوينو:

1- بعد أن تنتهي من كتابة الكود البرمجي – اضغط على زر تأكيد Verify في شريط الاوامر السريعة في أعلي الـ IDE ثم انتظر حتى تظهر رسالة في الاسفل تخبرك بأن البرنامج قد انتهي من تجهيز الكود ليتم رفعة على بوردة الأردوينو والشكل (13) يوضح ذلك .

الشكل (13)

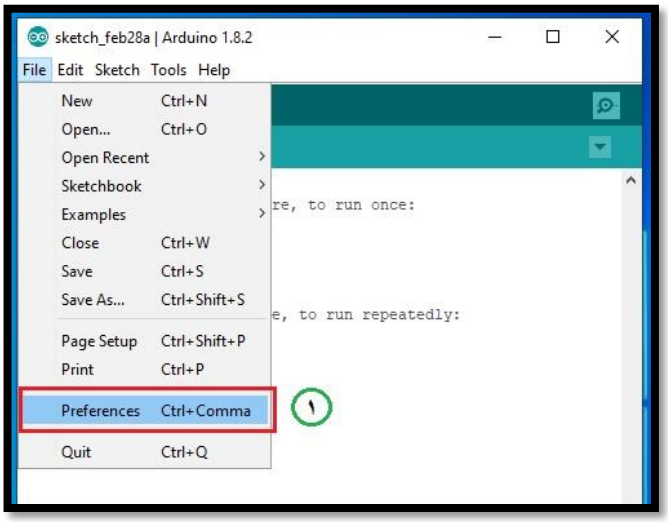
2- تأكد أن كابل الأردوينو متصل بالحاسب ثم قم بالضغط على زر رفع Upload وذلك حتى يتم بدأ عملية تحميل البرنامج الى المتحكمة الدقيقة وانتظر عدة ثواني ستلاحظ عندها أن الـ leds الموجودة على البوردة (مكتوبة بجانبها TX/RX) بدأت تضئ بسرعة عالية ثم تقف قليل وهذه الإضاءة تعني أن البورده بدأت تستقبل بيانات البرنامج الذي قمنا بكتابته وبذلك أصبحت البوردة جهازه للعمل والشكل (14) يوضح ذلك .



الشكل (14) يوضح خطوات Upload

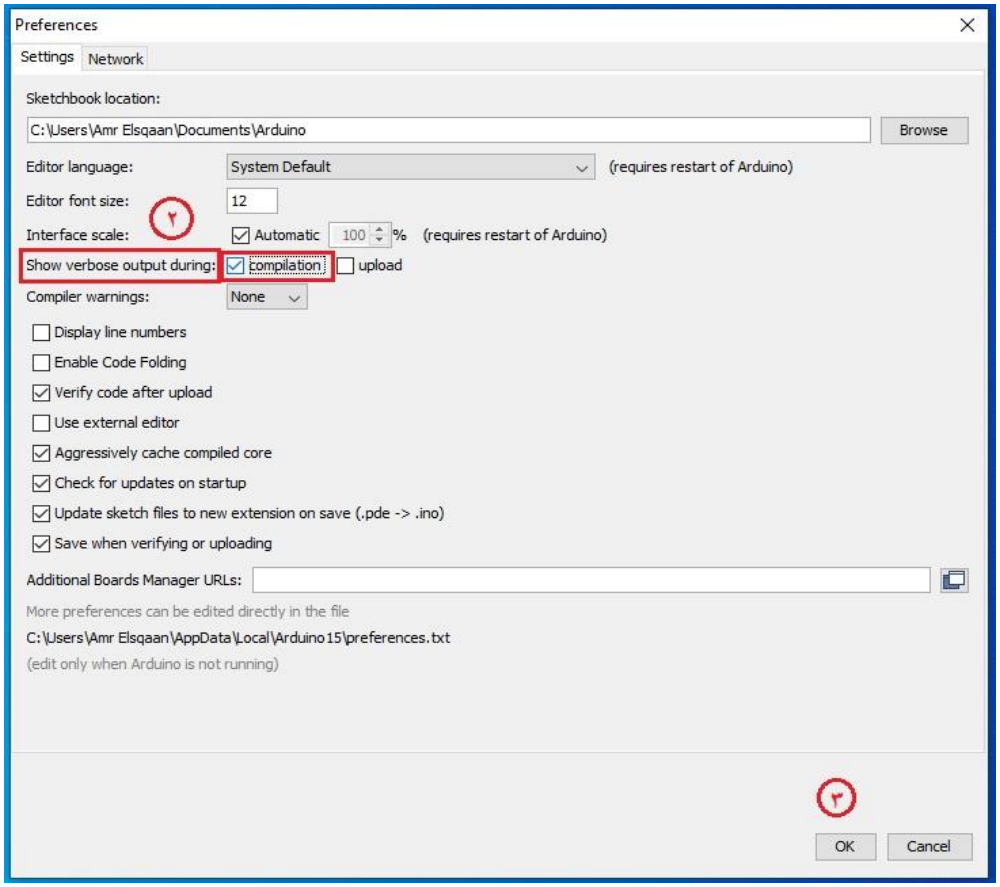
ثالثا : نقل الملف التنفيذي على لوحة الأردوينو في برنامج المحاكاة بروتس

1- بعد الانتهاء من كتابة البرنامج في البيئة التطويرية وتصحيحه اذهب إلى قائمة ملف (File) واختر الخيار preferences الشكل (15) يوضح ذلك .



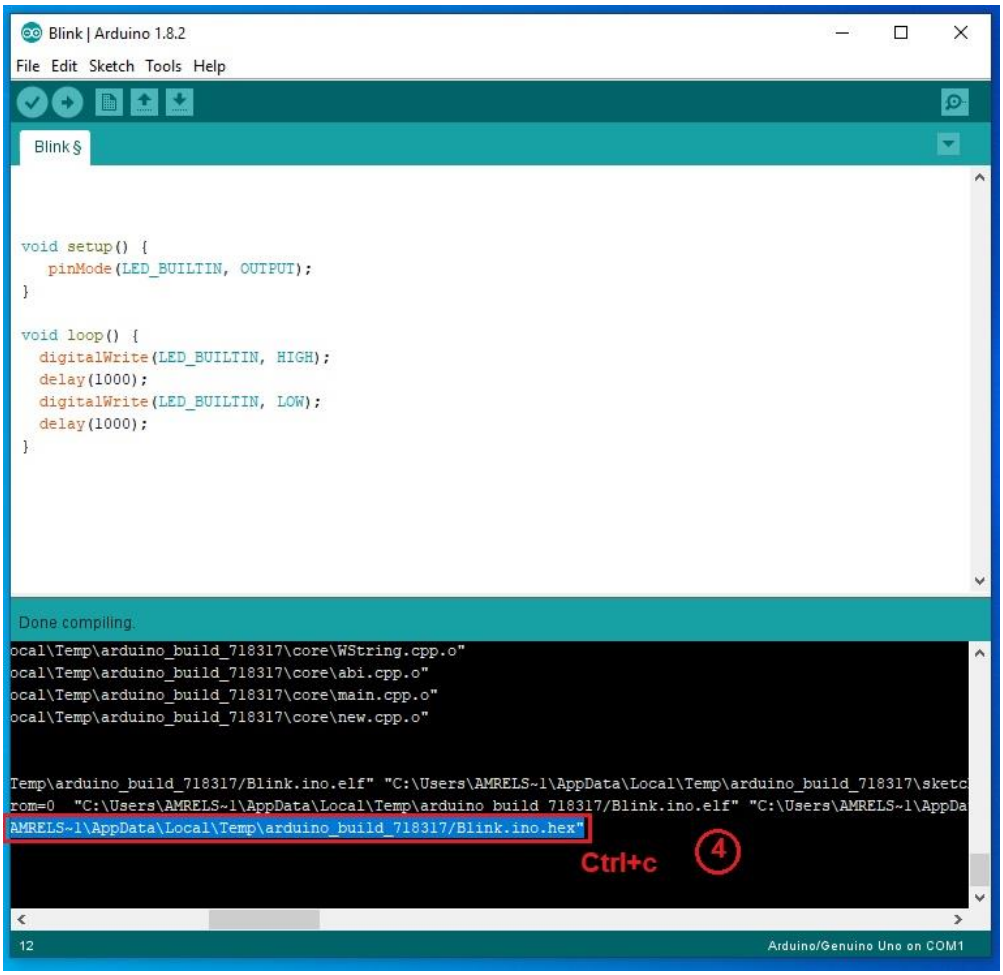
الشكل (15)

2- عند Show verbose output during ضع علامة صح على اختيار compilation ثم اضغط Ok الشكل (16) يوضح ذلك .

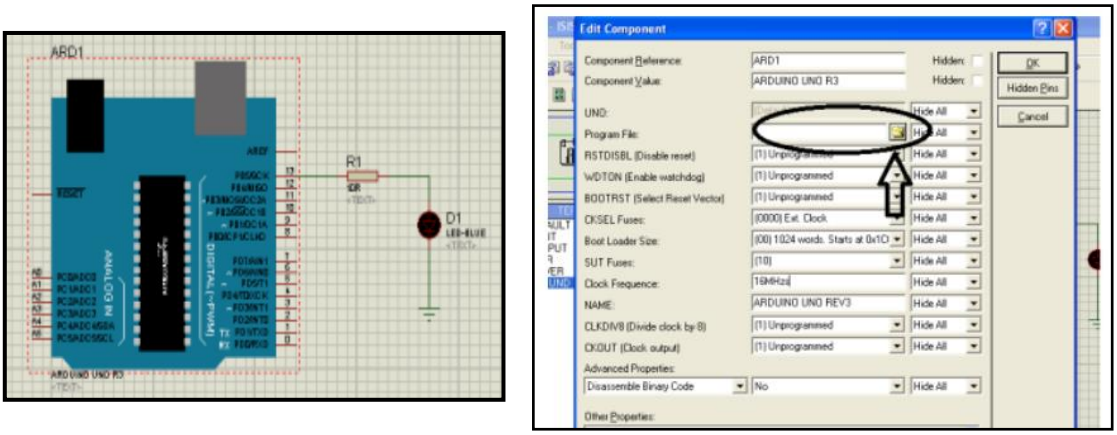


الشكل (16)

3- سنلاحظ ظهور شاشة سوداء أسفل البرنامج لكتابة التنبيهات والاخطاء البرمجية و سوف نلاحظ مسار ملف الـ hex ناتج عملية Compile . ننسخ مسار الملف ومن ثم قم بالضغط على زر Control +C من لوحة المفاتيح



4- بعد ذلك نضغط على المتحكمة في بوردة الأردوينو ببرنامج البروتس فتظهر الواجهة التالية ونلصق مسار ملف hex لينتقل البرنامج الى المتحكمة وتصبح جاهزة للعمل والشكل (17) يوضح ذلك .



الشكل (17) يوضح خطوات رفع ملف hex لمتحكمة الاردوينو ببرنامج البروتس

المخرج التعليمي (1)	ينفذ منظومة تحكم تعمل بالأردوينو باستخدام برامج المحاكاة		
معيـــــار الأداء (2) معيـــــار الأداء (3)	يكتب برنامج بلغة الأردوينو وفقا للتطبيق المطلوب . يختبر منظومة التحكم وفقا لمخطط الدائرة المعطاة		
زمن التنفيـــــــــذ		أدلة التعليم	مرفق 2 – مرفق 3
معلومات إثرائيـــــه	مجموع من الفيديوهات التي تتناول برمجة الأردوينو		

البرنامج (الكود البرمجي) :

الكود البرمجي : هي نصوص مكتوبة تم كتابتها من قبل مبرمجين لتقوم بعمل معين ، وهذه النصوص يطلق عليها لغة البرمجة وهي عبارة عن توجيهات وتعليمات يفهما المتحكم الدقيق ليقوم بتنفيذها .

والمتحكم الدقيق لا يفهم لغة البرمجة بشكل مباشر ، فهو لا يفهم إلا لغة واحدة وهي اللغة الثنائية Binary ، لذلك عادة ما يتم ترجمة لغات البرمجة إلى اللغة الثنائية عبر برنامج يسمى Compiler .

وأساسيات البرمجة تكاد تكون مشتركة في أغلب اللغات علي سبيل المثال (الثوابت – المتغيرات – حلقات loops – الجمل الشرطية – الدوال ...) وسوف نتناول تلك الأساسيات والتي سوف نحتاجها للتعامل مع برمجة الأردوينو .

التركيب البنائي للبرنامج (الكود البرمجي) في الأردوينو :

```
// Example_2_Blinking_with_Switch

const int ledPin = 13;
const int buttonPin = 2;
int val;

void setup ( )

{
 pinMode(ledPin, OUTPUT);
 pinMode(buttonPin, INPUT);

}

void loop()

{
 val = digitalRead(buttonPin);

 if (val == HIGH)

   {
 digitalWrite(ledPin, HIGH);
 delay(1000);
 digitalWrite(ledPin, LOW);
 delay(1000);
    }

 else {digitalWrite(ledPin, LOW); } }
```

المرحله الاولى: تعريف المتغيرات و الثوابت

المرحله الثانيه: تعريف و تحديد المداخل و المخارج

المرحله الثالثه: كتابه اوامر البرنامج الرئيسيه

الشكل (18)

المرحلة الاولي : في هذه المنطقة من بنية البرنامج يتم تعريف الثوابت والمتغيرات العامة أما تعريف المتغيرات المحلية فيتم تعرفيها في المرحلة الثالثة (سوف يتم تناول أنواع المتغيرات بالتفصيل) كذلك يمكن في هذه المنطقة كتابة التعليقات (الملاحظات) على البرنامج ويقوم بكتابتها المبرمج بهدف التسهيل على من يتعامل مع البرنامج بعد ذلك ومن التعليقات أو الملاحظات التي تكتب علي سبيل المثال الهدف من البرنامج شرح الكود البرمجي أو شرح احد الاسطر البرمجية ...

ويكتب التعليق بشكل معين حتى يفهم Compiler أن هذا النص مجرد ملاحظات وليس أمر برمجي والشكل (19) يوضع صور كتابة التعليقات والجدير بالذكر ان التعليقات تكتب في أي جزء في البرنامج وليس شرطا أن تكون في المرحلة الاولي فقط ويمكن أن تكتب التعليقات باللغة الإنجليزية أو العربية .

الشكل (19)

المرحلة الثانية : تبدء هذه المرحلة بجملة Void setup() يأتي بعدها قوسين متعرجين يتم ما بين القوسين تعريف وتحديد المداخل والمخارج للوحة الأردوينو وكذلك كتابة الأوامر أو الاسطر البرمجية التي يتم تنفيذها مرة واحدة فقط أثناء عمل البرنامج والشكل (20) يوضح ذلك .

الشكل (20)

المرحلة الثالثة : تبدء هذه المرحلة بجملة () Void loop يأتي بعدها قوسين متعرجين يتم ما بين القوسين كتابة أوامر البرنامج الرئيسية والتي تتكرر طوال فترة تشغيل البرنامج .

الاخطاء البرمجية:

يمكن تقسيم الأخطاء البرمجية إلى ثلاث أنواع :

- ☒ أخطاء قبل تنفيذ البرنامج ويكتشفها المترجم (Compiler) وهى الاخطاء الناتجة من قيام المبرمج بكتابة قواعد اللغة بطريقة خاطئة . مثل الفاصلة (;) بدلا من الفاصلة (;) في نهاية السطر البرمجي او يكتب كلمة محجوزة في اللغة بطريقة خاطئة مثل كتابة نوع البيانات Int بدلا من int .
- ☒ أخطاء تحدث أثناء تنفيذ البرنامج ولايمكن اكتشافها عن طريق المترجم Compiler . مثل تعريف متغير من النوع int يقوم باستقبال رقم صحيح من المستخدم الا ان المستخدم يقوم بإدخال رقم عشري بالخطأ . وهذه الاخطاء يجب على المبرمج عمل معالجة لها .
- ☒ أخطاء تحدث بعد التنفيذ البرنامج وتسمي بالأخطاء المنطقية او بمعنى اخر هو خطا فى نتيجة او مخرجات البرنامج . يتم اكتشافها بعد التنفيذ عند عمل test للبرنامج لايمكن اكتشافها عن طريق المترجم Compiler مثل حساب مساحة مربع بطريقة خاطئة .

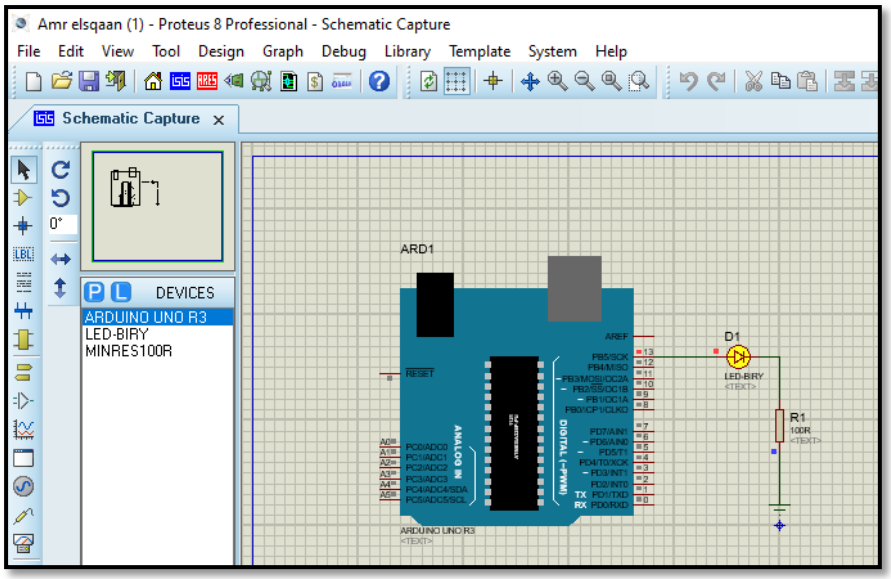
اسم التطبيـــــــق :	التحكم في إضاءة led		
زمن التطبيـــــــق :		رقم التطبيـق :	1
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno	4- أسلاك توصيل 5- مقاومة 100 اوم 6- دايود ضوئي LED 7- جهاز آفوميتر	

الهدف من التطبيق :

تطبيق ما تم دراسته من تجهيز بيئة IED ، ونقل الملف التنفيذي على لوحة الأردوينو ، والتركيب البنائي للبرنامج وطريقة كتابته .

المطلوب من التطبيق : إضاءة وإطفاء موحد ضوئي لعدد من المرات

أ- المخطط باستخدام برنامج المحاكاة Proteus :



شكل (21)

1- افتح برنامج Proteus وأحضر كلا من لوحة الأردوينو أنو - موحد ضوئي LED – مقاومة كربونية من مكتبة العناصر

2- أحضر الأرضي وقم بتوصيل الدائرة كما هو موضح بالشكل (21).

ب- كتابة الكود البرمجي لتشغيل الدائرة باستخدام برنامج Arduino IED :

```
1  void setup()
2  {
3    pinMode(13, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(13, HIGH);
9    delay(1000); // Wait for 1000 millisecond(s)
10   digitalWrite(13, LOW);
11   delay(1000); // Wait for 1000 millisecond(s)
12 }
```

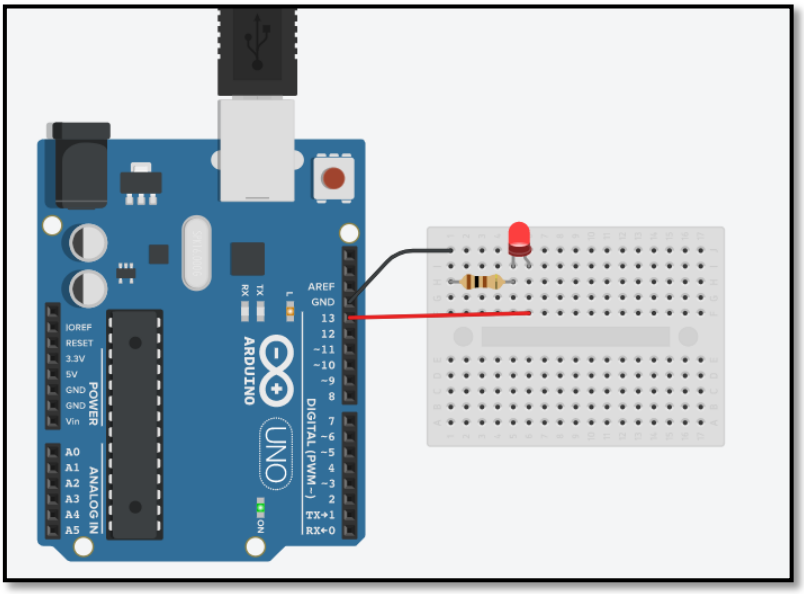
شكل (22)

شرح الكود :

م	الوصف
1	نستدعي دالة الإعدادات
2	فتح قوس دالة الاعدادات
3	تعريف الطرف 13 بأنه خرج

م	الوصف
4	غلق قوس دالة الاعدادات
6	نستدعي دالة حلقة البرنامج اللانهائية
7	فتح قوس دالة حلقة البرنامج
8	أضاءه LED
9	الانتظار لمدة ثانية
10	غلق إضاءة LED
11	الانتظار لمدة ثانية
12	غلق قوس دالة حلقة البرنامج

جـ - توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard :



شكل (23)

في التطبيق السابق . عند كتابتك الكود البرمجي له غالبا ما قمت بخطأ برمجي حدد مع معلمك ما نوع الخطأ البرمجي

تمرين (1)

باستخدام برنامج المحاكاة :

1- أضف عدد 3 موحدات ضوئية على المخارج (12و 11 و10) ثم عدل الكود بحيث يضئ الموحد الضوئي على المخرج 13 ثم يطفأ ليضيء الموحد الضوئي على المخرج 12 ثم يطفأ وهكذا ..

2- بعد تنفيذ التمرين والتأكد من سلامة الكود البرمجي قم بتعديل الأزمنة من 1 ثانية إلى 500 ملي ثانية ثم 250ملي ثانية وفي كل مرة عدل الكود

(الحل في دليل حل التمارين بنهاية الوحدة)

المتغيرات Variable :

المتغيرات هي عبارة عن حاويات لها حجم معين تقوم بتخزين البيانات، وتعد من أساسيات البرمجة ولا يمكن لبرنامج أن يعمل من دونها، وتتكون من عدة انواع لكل نوع حجم مخصص له في الذاكرة و لا يمكن أن يتجاوز هذا الحجم والجدول التالي يوضح أنوع المتغيرات في برمجة الأردوينو .

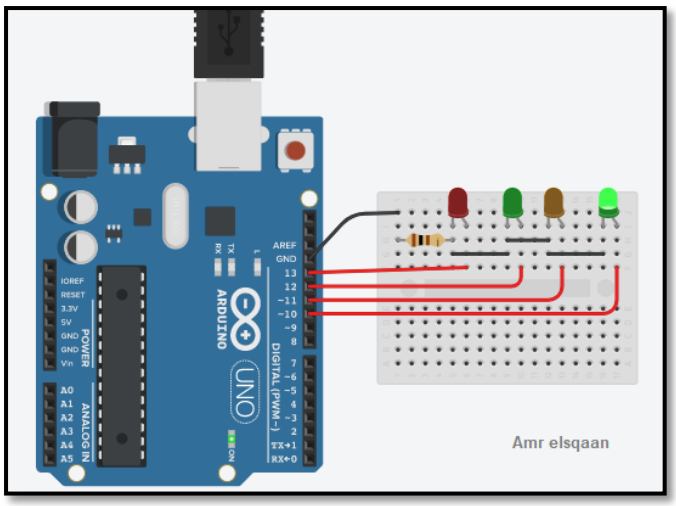
م	اسم المتغير	الوصف
1	Boolean	يمكن لهذا النوع من المتغيرات الاحتفاظ بواحدة من قيمتين فقط و هما true أو false. يستخدم هذا النوع من المتغيرات لحفظ نتائج العمليات المنطقية. يحتاج هذا المتغير إلى بايت واحد من الذاكرة. مثال boolean result = true;
2	Char	يمكن لهذا النوع من المتغيرات الاحتفاظ بحرف واحد. ويحاج هذا النوع من المتغيرات إلى بايت واحد من الذاكرة .مثال char letter = 'A';
3	Byte	يمكن لهذا النوع من المتغيرات خزن بايت واحد من الأرقام الموجبة. أي انه يمكن تخزين أي قيمة تتراوح من 0 لغاية 255. يقوم هذا المتغير بحجز بايت واحد من الذاكرة. مثال byte m = 180;
4	int	يمكن لهذا النوع من المتغيرات خزن الأرقام الموجبة و السالبة. و يختلف حجم هذا المتغير من لوح أردوينو إلى اخر فمثلا على لوح Ardoino Due يكون حجم المتغير من هذا النوع 2 بايت (16 بت) يتم حجز احدها للإشارة فيتبقى 15 بت. ويعتبر هذا النوع من المتغيرات الأكثر استخداماً في برامج الآردوينو. مثال . int counter = 15;
5	Word	يقوم هذا المتغير بحجز 2 بايت من الذاكرة في الآردوينو Uno و ATMEGA و 4 بايت من الذاكرة في الأردوينو Due و Zero. يمكن خزن أرقام موجبة فقط في هذا النوع من المتغيرات. مثال word level = 800;
6	Long	يقوم هذا النوع من المتغيرات بحجز 4 بايت من الذاكرة, يستخدم احد البتات للإشارة و 31 بت لخزن القيمة. مثال. long value = 4588642;

م	اسم المتغير	الوصف
7	Short	يقوم هذا النوع من المتغيرات بحجز 2 بايت من الذاكرة يستخدم احد البتات فيها للإشارة. مثال . short tlength = 58
8	Float	يمكن لهذا النوع من المتغيرات خزن أرقام تحتوي على فاصلة عشرية. أي انه يمكن استخدامه مع الأرقام غير الصحيحة و يقوم هذا النوع من المتغيرات بحجز 4 بايت من الذاكرة. مثال float pi = 3.14
9	Double	يتطابق هذا النوع من المتغيرات مع الـ float في الآردوينو Uno و ATMEGA في حين يكون اكثر دقة في الأردوينو Due لكونه يحجز 8 بايت من الذاكرة فيها.

والان وبعد تنفيذ التمرين (1) وتغيير الأزمنة من 1 ثانية إلى 500 ملي ثانية . كم تعديل قمت به في الاسطر البرمجية ؟ هل تتذكر كم المجهود في تعديل الكود لتغيير زمن التاخير ;delay (500) ؟

الحل في استخدام المتغيرات وفيما يلي أعد كتابه الكود البرمجي باستخدام المتغيرات .

☒ توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard وكتابة الكود البرمجي :




```
int d=250;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay (d);
  digitalWrite(13, LOW);
  delay (d);
  digitalWrite(12, HIGH);
  delay (d);
  digitalWrite(12, LOW);
  delay (d);
  digitalWrite(11, HIGH);
  delay (d);
  digitalWrite(11, LOW);
  delay (d);
  digitalWrite(10, HIGH);
  delay (d);
  digitalWrite(10, LOW);
  delay (d);
}
```

➢ في السطر (1) :

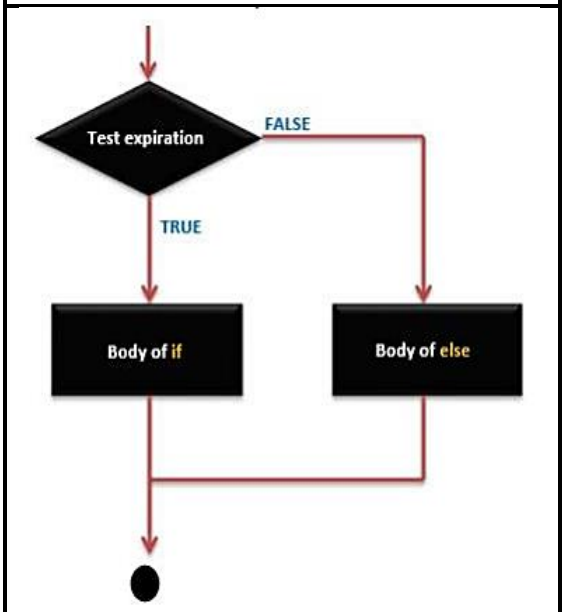
- تم وضع متغير من نوع int أسمة d وتم وضع قيمة له 250 .
- وبدلا من وضع قيمة للامر delay يوضع المتغير d بدلا من القيمة .
- في كل مرة نريد نغير قيمة delay نغير في قيمة المتغير d فقط .

الشروط Conditions :

تعتبر العبارات الشرطية في البرنامج من الامور الاساسية في البرمجة وتمتلك لغة Arduino C نوعين من العبارات الشرطية هما بنية (if-else) وبنية (switch- case)

أ- بنية (if-else) :

```
if (expression)
  {
    Block of statements;
  }
else
  {
    Block of statements;
  }
```



- مثال :

```
int  A = 5 ;
int  B=  9 ;
Void setup ()
{
}
Void loop ()
{
1 ➡ if (A > B)
      {
        A++;
      }
2 ➡ else
      {
        B -= A;
      }
}
```

1- اذا تحقق الشرط بعد جملة if :
- سوف يتم تنفيذ ما بين الاقواس { } التي تلي جملة if

2- اذا لم يتحقق الشرط بعد جملة if
- سوف يتم تنفيذ ما بين الاقواس { } التي تلي جملة else

اسم التطبيـــــــق :	إضاءة led ليعمل فقط عند ضغط مفتاح من نوع Push button		
زمن التطبيـــــــق :		رقم التطبيـق :	2
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- مقاومة 10kΩ 5- أسلاك توصيل	6- مقاومة 560Ω 7- دايود ضوئي LED 8- جهاز آفوميتر 9- مفتاح Push button	

الهدف من التطبيق :

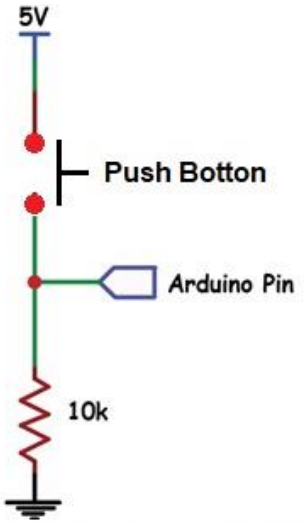
تطبيق ما تم دراسته بجملة (if) بالإضافة إلى استخدام مفتاح Push button وهو من أشهر المفاتيح المستخدمة في الأردوينو كذلك استخدام أحد الأطراف الرقمية كمدخل .

تمهيد : طرق توصيل مفتاح Push button بالأردوينو :

هناك طريقتان لتوصيل مفتاح Push button إلى لوحة الأردوينو وهما :

1- مقاومة الخفض Pull down Resistor

في هذه الطريقة من توصيل المفتاح . عند الضغط على المفتاح يصل الى دخل الأردوينو 5 فولت ليتعرف الأردوينو على انه "1" منطقي ، أما في حالة تحرير المفتاح (عدم الضغط علية) فسوف يصل الأرضي الى مدخل الأردوينو ليتعرف عليه أنه "0" منطقي . وتوضع المقاومة 10 k حتى لا يحدث Short عند الضغط على المفتاح .



عند الضغط على المفتاح يمر ٥ فولت الى الأردوينو

Pull Down Resistor

2- مقاومة الرفع Pull up Resistor

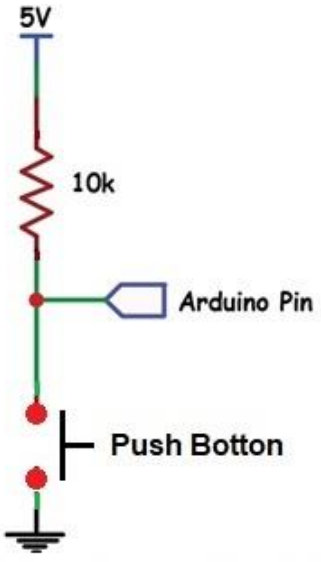
في هذه الطريقة من توصيل المفتاح . عند الضغط على المفتاح يصل الى دخل الأردوينو صفر فولت ليتعرف الأردوينو على انه "0" منطقي ، أما في حالة تحرير المفتاح (عدم الضغط علية) فسوف يصل 5 فولت الى مدخل الأردوينو ليتعرف عليه أنه "1" منطقي .

الجدير بالذكر أنه يمكن الاستغناء عن توصيل المقاومة وتوصيل المفتاح مباشرة بشرط استخدام الكود البرمجي التالي :

```
PinMode(10,INPUT_PULLUP);
```

بدلا من :

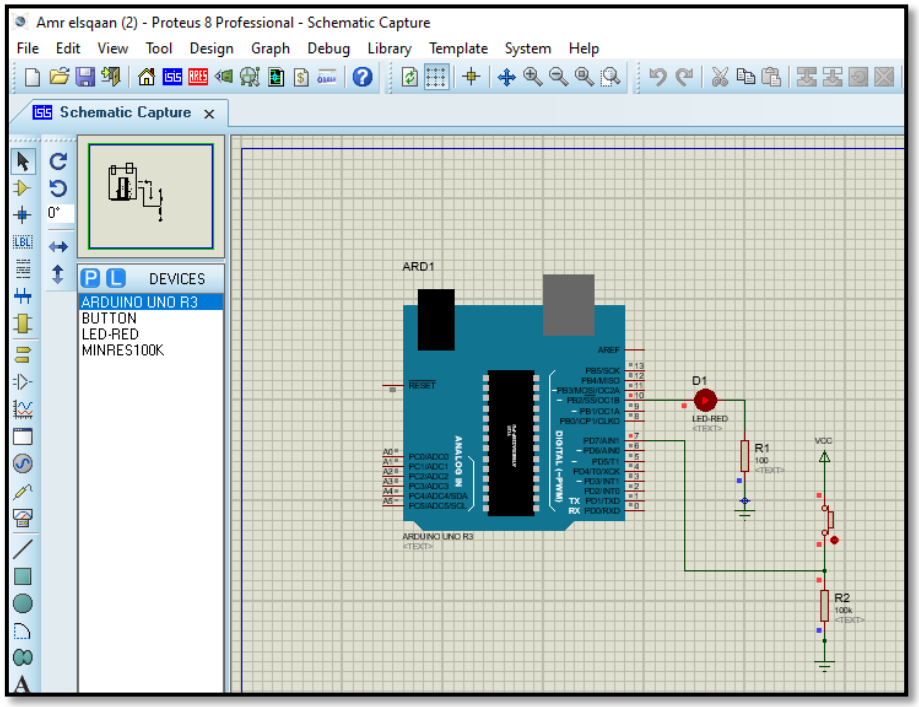
```
PinMode(10,INPUT);
```



عند الضغط على المفتاح يصل صفر فولت الى مدخل الأردوينو

Pull up Resistor

أ- المخطط باستخدام برنامج المحاكاة Proteus :



الشكل (24)

1- افتح برنامج Proteus وأحضر كلا من لوحة الأردوينو أنو - موحد ضوئي LED – مقاومة كربونية 100 أوم – مقاومة 100k - مفتاح من مكتبة العناصر

2- أحضر الأرضي وقم بتوصيل الدائرة كما هو موضح بالشكل (24).

ب- كتابة الكود البرمجي لتشغيل الدائرة باستخدام برنامج Arduino IED :

شرح الكود :

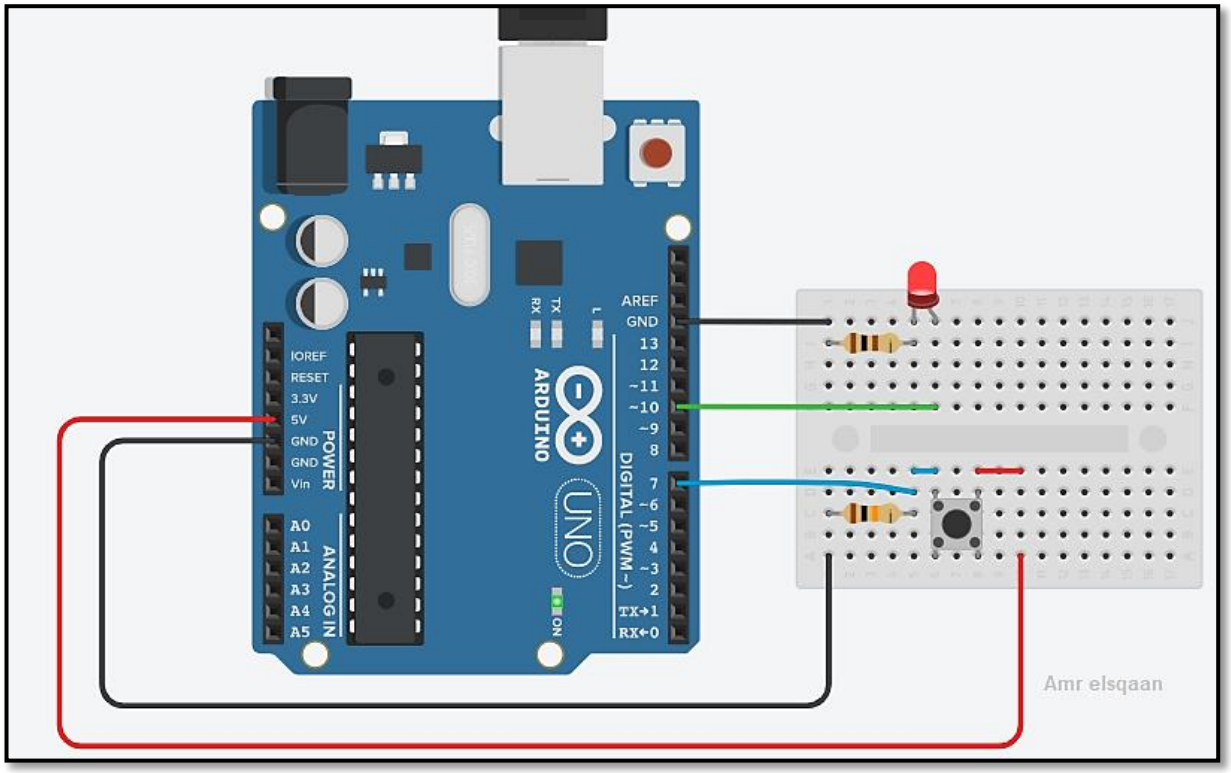
م	الوصف
1	نستدعي دالة الإعدادات
3	تعريف الطرف 10 على انه خرج
4	تعريف الطرف 7 على أنه دخل
7	نستدعي دالة حلقة البرنامج اللانهائية
9	استخدام جملة IF إذا الدخل 7 = HIGH نفذ التالي
10	انتظر ثانية
11	أضاء LED
12	انتظر ثانية
13	أغلق LED

```
1  void setup()
2  {
3    pinMode(10, OUTPUT);
4    pinMode(7,INPUT);
5  }
6
7  void loop()
8  {
9    if(digitalRead(7)==HIGH)
10   {
11   digitalWrite(10, HIGH);
12   delay(1000);
13   digitalWrite(10, LOW);
14   delay(1000);
15   }
16   else
17    digitalWrite(10, LOW);
18 }
```

تابع شرح الكود :

م	الوصف
14	انتظر ثانية
15	أغلق قوس جملة IF
16	أذا لم يتحقق شرط IF
17	أغلق أضاءه LED

جـ - توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard :



تمرين (2)

باستخدام برنامج المحاكاة ولوحة الاردوينو :

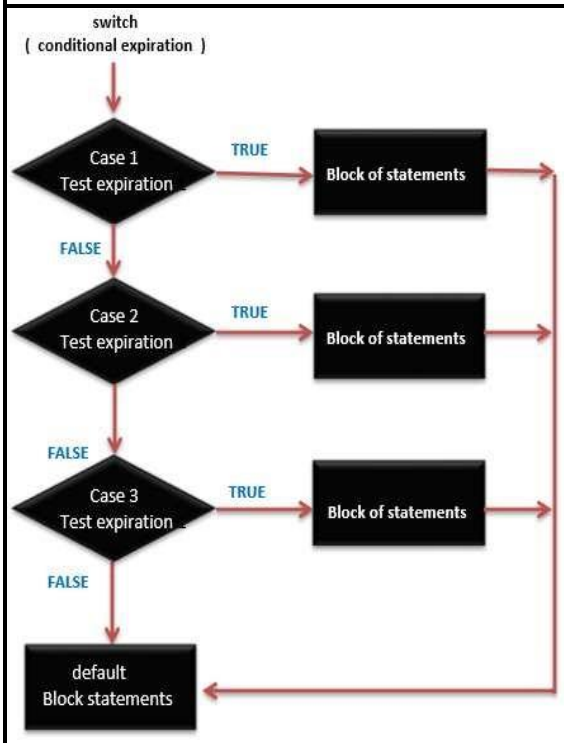
1- اجري التعديل المناسب على التطبيق السابق ليصبح دائرة المفتاح تعمل بنوع مقاومة الرفع Pull up Resistor ثم عدل الكود بحيث يضئ LED مباشرة وعند الضغط على المفتاح لا يعمل LED ثم اشرح الكود الذي قمت بتعديله

(الحل في دليل حل التمارين بنهاية الوحدة)

ب- بنية (switch- case) :

على غرار عبارات if ، يتحكم switch ... case في تدفق البرامج ، وهى مجموعة عبارات شرطية (case) ويقارن القيمة عند كل (case) مع المتغير في (switch) ويتم التحقق من الشروط وقت التنفيذ فاذا لم يتحقق الشرط الاول ينتقل الى (case) الثانية فاذا لم يتحقق الشرط الثاني ينتقل الى (case) الثالثة حتى اخيرا يصل الى الشرط الذى يتحقق وذا تحقق واحد من (case) سوف ينفذ ما في داخلها ويهمل البقية واذا لم ينفذ أي واحد منهن سوف يتجه لينفذ ما في داخل (default) .

```
switch (variable)
{
    case label:
   // statements
       break;
    }
  case label:
    {
 // statements
    break;
     }
  default:
    {
     // statements
      break;
```



- مثال :

المتغير data يمكن أن يأخذ قيم مختلفة وعلى سبيل المثال (0,1,2) وحسب القيمة التي اخذها يذهب الى case المحددة لينفذ الجمل البرمجية .

```
Int data = 2;

switch (data)
}
case 0:
// ……;
// ……;
 break;
case 1:
// ……;
// ……;
 break;
case 2:
// ……;
// ……;
 break;
default:
//……;
}
```

الصيغة العامة للأمر : tone يأتي الامر في صورتين كما يلي :

Tone (10,400); // يعمل على اصدار صوت من المنفذ 10 بتردد 400 هيرتز
noTone(10); // يعمل على ايقاف النغمة على المنفذ 10
	1

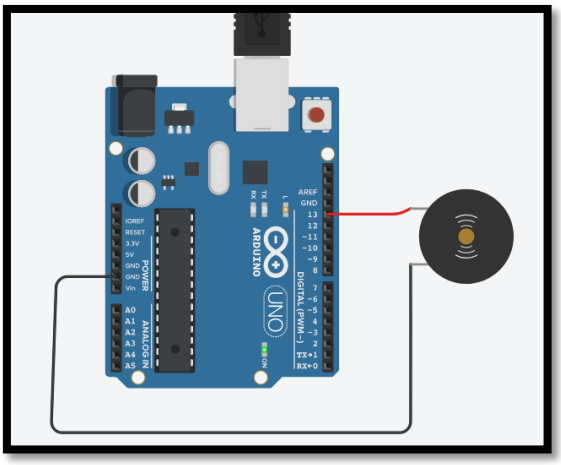
Tone(10,400,500); // يمثل الرقم الثالث فترة تشغيل النغمة وتكون بالملي ثانية	2

اسم التطبيـــــــق :	اصدار صوت باستخدام الامر tone		
زمن التطبيـــــــق :		رقم التطبيـق :	3
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- أسلاك توصيل	5- جهاز آفوميتر 6- مفتاح Push button 7- سماعة BUZZER	

الهدف من التطبيق :

استخدام الامر tone لتوليد النغمات حسب التردد كذلك استخدام أحد الأطراف الرقمية كمدخل .

أ- توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard :



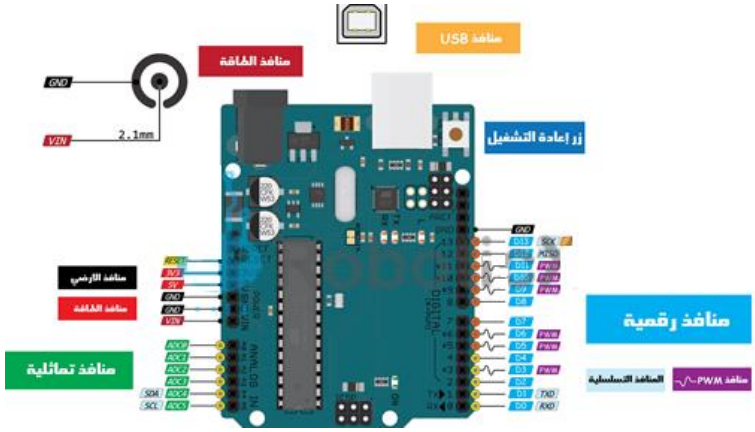
شرح الكود :

م	الوصف
1	أنشى متغير نوع INT واسمة LS وضع له قيمة 13
3	استدعي دالة الاعدادات
4	استخدم أمر tone لمنفذ 13 بتردد 250 لفتره زمنية 500 ملي ثانية
7	أغلق قوس دالة الاعدادات

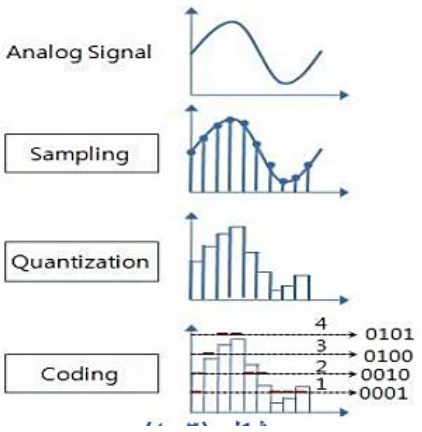
```
1  int ls=13;
2
3  void setup()
4  {
5
6    tone( ls,250,500);
7  }
8
9  void loop()
10 {
11
12 }
```

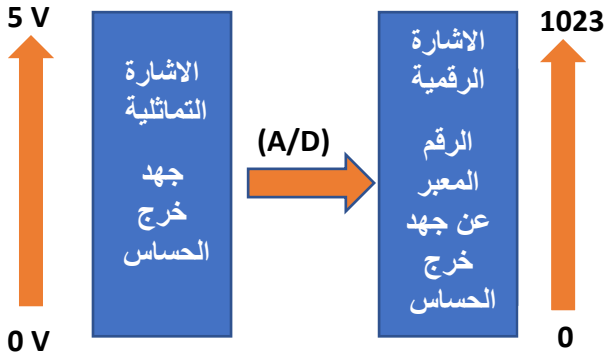
المداخل التناظرية Analog Inputs

يوجد 6 منافذ دخل تماثلية ومعنونه من A0 إلى A5 بشكل افتراضي وتستطيع هذه المداخل قياس جهد من 0 إلى 5 فولت ولكي يفهم الأردوينو الاشارة التماثلية نقوم باستدعاء دالة مهمة وهي analogRead(pin); ويتم وضع رقم الـ pin حسب المدخل المستخدم A5 to A0 هذه الدالة تقوم بعمل تحويل للإشارة التماثلية إلى إشارة رقمية .



و ترجع أهمية استخدام المداخل التماثلية إلى أن حواسنا كبشر (السمع – البصر- اللمس- الشم) لا تتعامل إلا مع الاشارات التماثلية وبالتالي فمعظم الحساسات تحول الكمية الفيزيائية (حرارة – ضوء- صوت - ..الخ) الى جهد او مقاومة لذا يوجد في المتحكم Atmega المستخدم في الأردوينو محول من تماثلي الى رقمي (A\D) ليقوم بتحويل الجهد التناظري القادم من الحساس والمعبر عن القيمة الفيزيائية (صوت – ضوء- حرارة - ..الخ) الى اشاره رقمية عبارة عن رقم بحيث يناظر جهد 5 فولت (وهو أقص جهد يتعامل معه الأردوينو) رقم 1023 (وهو أقصى سعة تخزينية لـ A/D وهي 1024 = 2^{10} أي من 0 إلى 1023) ، فلو فرضنا على سبيل المثال أن حساس للحرارة متصل بالمدخل A0 قام بتحويل درجة الحراة فأعطى جهد 5 فولت فيقوم المحول التماثلي الى رقمي (A\D) بتحويل 5 فولت المعبرة عن درجة الحرارة الى رقم 1023 ، فلو فرضنا أن درجة الحرارة انخفضت ليعطي الحساس جهد 3 فولت فكم يكون القيمة الرقمية المناظرة ؟ بحسبة بسيطة تكون 613.8 ،وبذلك نكون حولنا الاشارة التماثلية الى اشارة رقمية يستطيع المتحكم التعامل معها والشكل التالي يوضح مراحل التحويل (A\D) .





اسم التطبيـــــــق :	التحكم في اضاءة LED عن طريق مستشعر لشدة الاضاءة LDR	
زمن التطبيـــــــق :		رقم التطبيـق : 4
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- أسلاك توصيل	5- جهاز آفوميتر 6- LDR 7- LED 8- مقاومة كربونية 10K,220

الهدف من التطبيق :

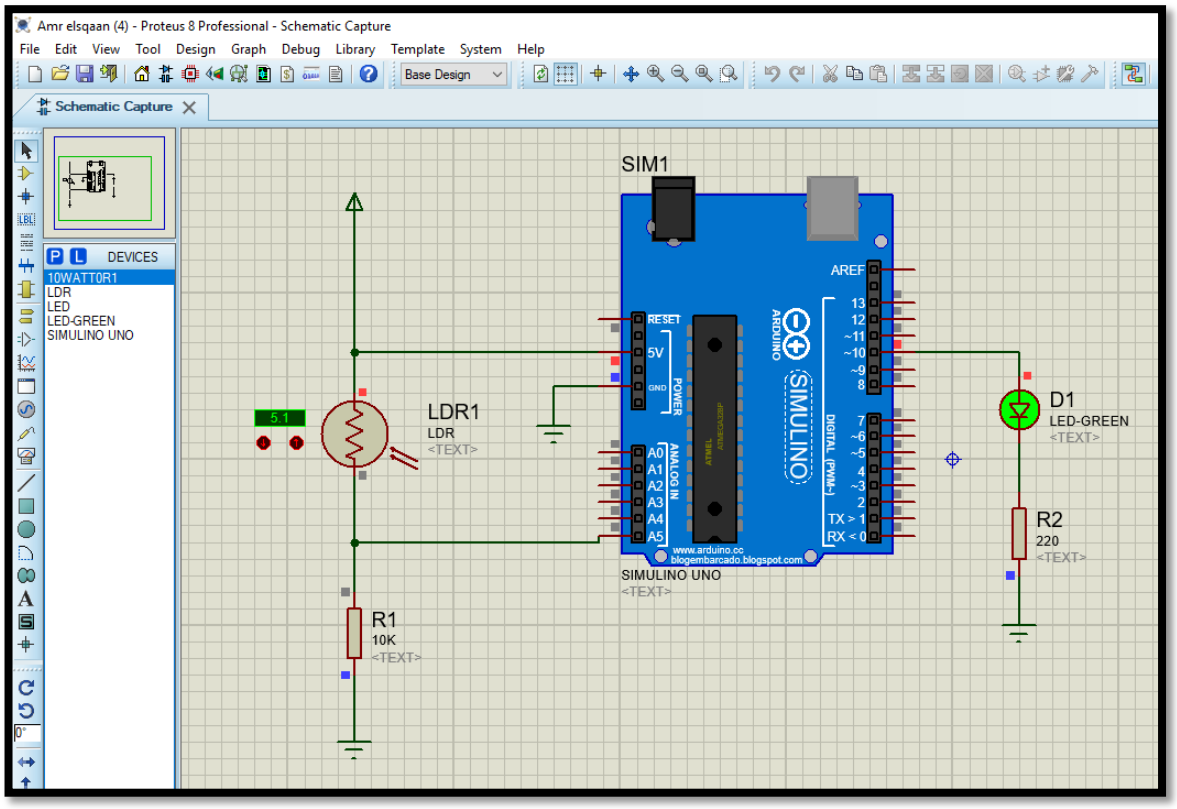
1- يهدف التطبيق الى التدريب على استخدام المداخل التماثلية للوحة الأردوينو

2- استخدام المقاومة الضوئية Sensor LDR مع لوحة الأردوينو لتحكم في أضاءه LED

تمهيد :

المقاومة الضوئية LDR هي عبارة عن عنصر إلكتروني يحتوى على مادة شبه موصلة حساسة للضوء وتكون قيمة هذه المقاومة بين طرفيها كبيرة جدا في الظلام التام وتقل تدريجا كلما سلط عليها الضوء بحيث تتناسب قيمة المقاومة عكسيا مع شدة الضوء الساقط على المقاومة .

أ- المخطط باستخدام برنامج المحاكاة Proteus :



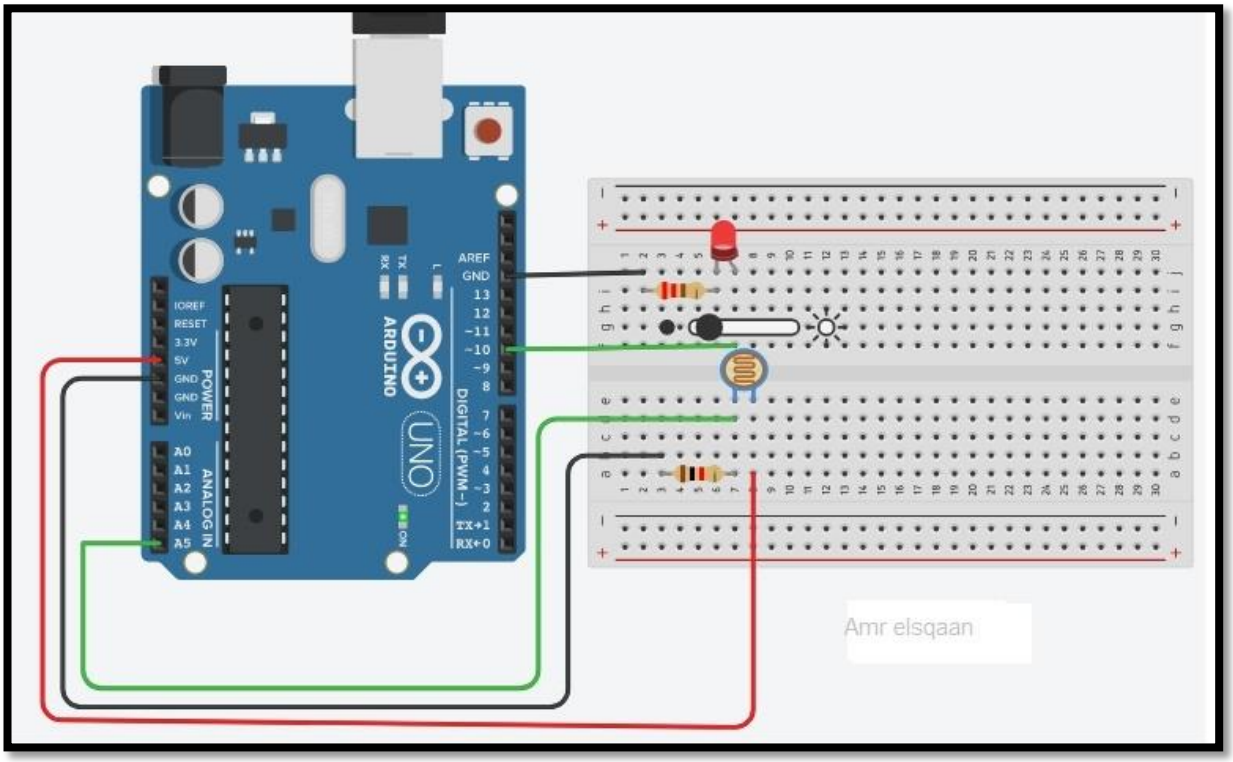
ب- كتابة الكود البرمجي لتشغيل الدائرة باستخدام برنامج Arduino IED:

```
int LED =10;
int LDR = A5;
int SensorReading =0 ;

void setup()
{
pinMode(LED,OUTPUT);

}

void loop()
{
 SensorReading = analogRead(LDR);
  if(SensorReading < 300 ){

      digitalWrite (LED,HIGH);
    }
  else  digitalWrite (LED,LOW);
  delay (1000);
}
```

شرح الكود :

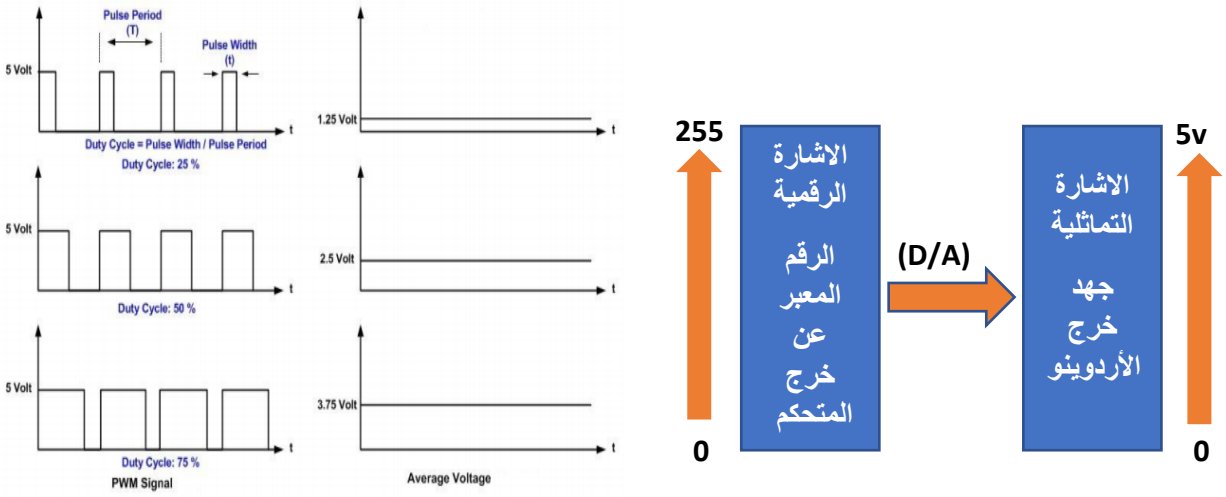
م	الوصف
1	أنشئ متغير واعطي له اسم LED وقيمة 10
2	أنشئ متغير واعطي له اسم LDR وقيمة A5
3	أنشئ متغير واعطي له اسم ..Sen وقيمة 0
7	أجعل المتغير LED والذي يمثل الطرف 10 كخرج
13	اقرأ قيمة الحساس على المدخل LDR وضعها في المتغير SensorReading
14	جملة if أذا كانت قيمة المتغير أقل من 300 نفذ ما يلي
16	أضئ LED
18	أذا لم يتحقق الشرط أطفأ LED
13	أنتظر ثانية

جـ - توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard :



المخارج التناظرية Analog Outputs

بوردة Arduino UNO بها 6 منافذ بهم هذه العالمة (~) يمكن استخدامهم كمخارج للإشارات الـ outputs analog أو مداخل ومخارج للإشارات الرقمية digital وهذه المنافذ هي (11,10,9,6,5,3) . وهذه المنافذ مزودة بخاصية تقوم بتعديل عرض النبضة وتسمي بـ(PWM) Modulation Width Pulse وهذه الطريقة يتم استخدامها للحصول على خرج تماثلي (فرق جهد متغير) بواسطة إشارة رقمية ويتم ذلك عن طريق التحكم فى زمن تشغيل الموجة (عرض الموجة) فإذا كان فرق الجهد على هيئة موجة مربعة قيمتها تتغير من 0V to 5V, و كانت الفترة التي يكون الجهد فيها 5V جهد التشغيل تمثل ربع الدورة الواحدة 25% , وكانت هذه الدورة تتكرر بنفس القيم سيكون عندها متوسط الجهد الناتج عبارة عن ربع الجهد الكلى 1.25V = (1/4) * 5 : وهكذا لوكان 5V جهد التشغيل يمثل نص الموجة 50% سيكون الجهد الناتج يمثل نص الجهد الكلي 2.5V=(1/2)*5 : وهكذا يمكننا التحكم في الجهد الناتج من الأردوينو كما هو موضح .



وبالتالي إذا اردنا الحصول على أي قيمة analog فإننا نقوم بتغيير عرض النبضة Width Pulse ويتم ذلك برمجيا باستخدام الامر analogWrite(pin,value); حيث يكتب مكان الـ pin رقم منفذ الخرج المستعمل في الدائرة, ونكتب مكان value قيمة عرض النبضة وهي تتراوح من 255: 0 .

وفي حالة استعمال المداخل والمخارج التماثلية هناك دالة تستخدم بكثرة بدلا من أن نقوم بالحسابات السابقة وهي

map(value, from low, to high, from low, to high)

value >> نكتب فيها قراءة المدخل التماثلى - Sensor .
from low,to high >> أقل قيمة وأعلى قيمة للـ sensor .
from low,to high >> أقل قيمة وأعلى قيمة للـخرج (المصباح مثلاً).

اسم التطبيـــــــق :	التحكم في شدة اضاءة LED عن طريق مستشعر لشدة الاضاءة LDR	
زمن التطبيـــــــق :		رقم التطبيـق : 5
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- أسلاك توصيل	5- جهاز آفوميتر 6- LDR 7- LED 8- مقاومة كربونية 220, 10K

الهدف من التطبيق :

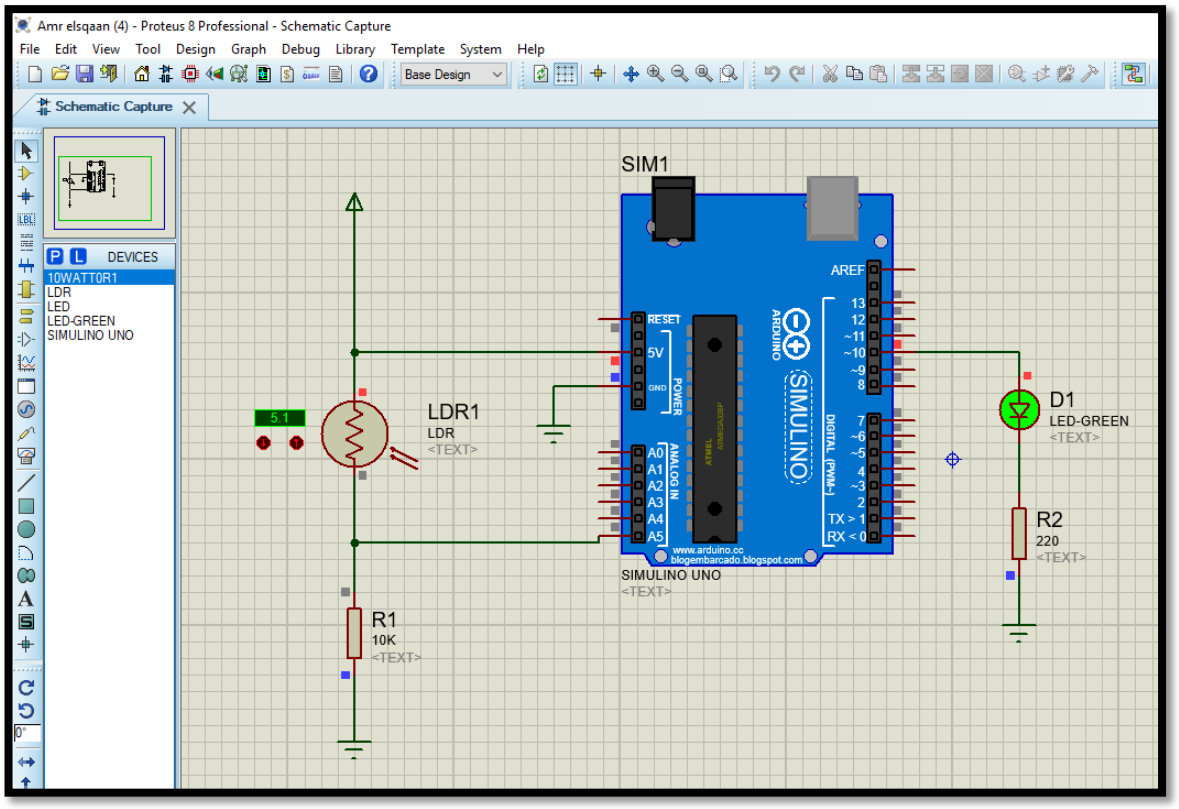
1- يهدف التطبيق الى التدريب على استخدام المداخل والمخارج التماثلية للوحة الأردوينو

2- استخدام المقاومة الضوئية Sensor LDR مع لوحة الأردوينو لتحكم في أضاءه LED

تمهيد :

تعتمد فكرة التطبيق على التحكم في شدة إضاءة LED عن طريق الضوء الساقط على المقاومة الضوئية LDR بحيث كلما قل الضوء الساقط على المقاومة الضوئية زاد شدة الضوء المنبعث من LED والعكس كذلك .

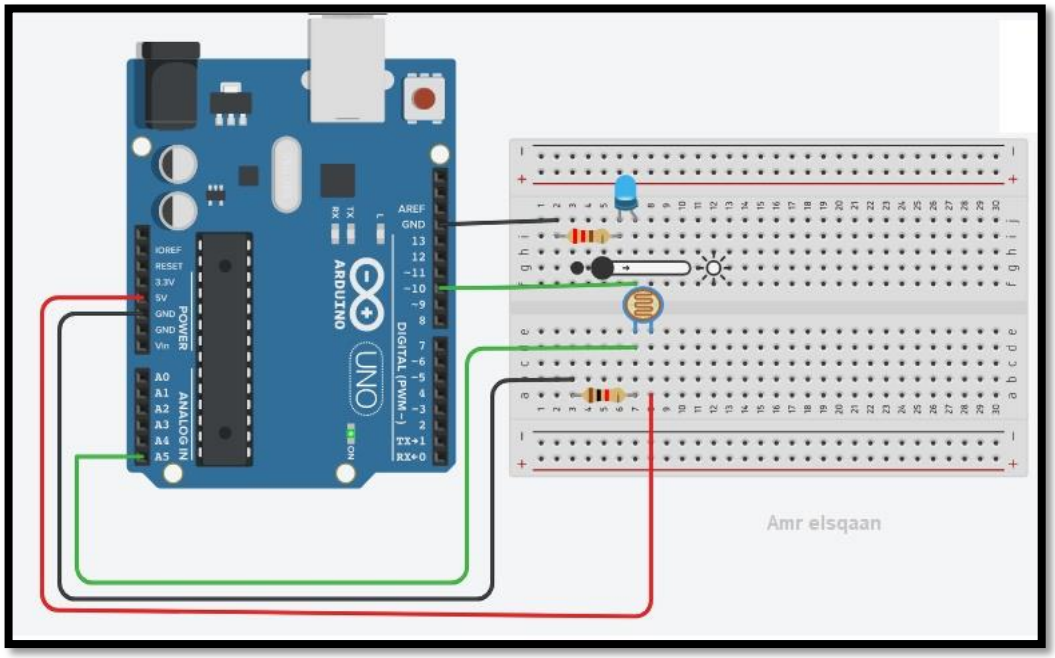
أ- المخطط باستخدام برنامج المحاكاة Proteus :



ب - كتابة الكود البرمجي لتشغيل الدائرة باستخدام برنامج Arduino IED:

```
int LED =10;
int LDR = A5;
int SensorReading =0 ;
int valMap = 0;

void setup()
{
pinMode(LED,OUTPUT);

}

void loop()
{
 SensorReading = analogRead(LDR);
 valMap = map(SensorReading ,0,800,255,0);
 analogWrite (LED,valMap);
}

```

ج- توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard :



شرح الكود :

م	الوصف
1	أنشئ متغير واعطي له اسم LED وقيمة 10
2	أنشئ متغير واعطي له اسم LDR وقيمة A5
3	أنشئ متغير واعطي له اسم .SensorReading وقيمة 0
4	أنشئ متغير واعطي له اسم valMap وقيمة 0
8	أجعل المتغير LED والذي يمثل الطرف 10 كخرج
12	نستدعي دالة حلقة البرنامج اللانهائية
13	فتح قوس دالة حلقة البرنامج
14	اقرأ قيمة الحساس وضع القيمة في المتغير SensorReading
15	استخدم الدالة map لحساب قراءة الحساس وضع نتيجة الحسابات في المتغير ValMap
16	أضئ LED بالقيمة المخزنة في المتغير ValMap
17	غلق قوس دالة حلقة البرنامج

عزيزي المتدرب:

تركيزك في التدريب يزيد من قدراتك المهارية ويقيك من المخاطر .

اسم التطبيـــــــق :	التحكم في درجة حرارة غرفة عن طريق حساس حرارة		
زمن التطبيـــــــق :		رقم التطبيـق :	6
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- أسلاك توصيل		5- جهاز آفوميتر 6- حساس حرارة LM35 7- LED

الهدف من التطبيق :

1- يهدف التطبيق الى التدريب على استخدام المداخل والمخارج التماثلية للوحة الأردوينو

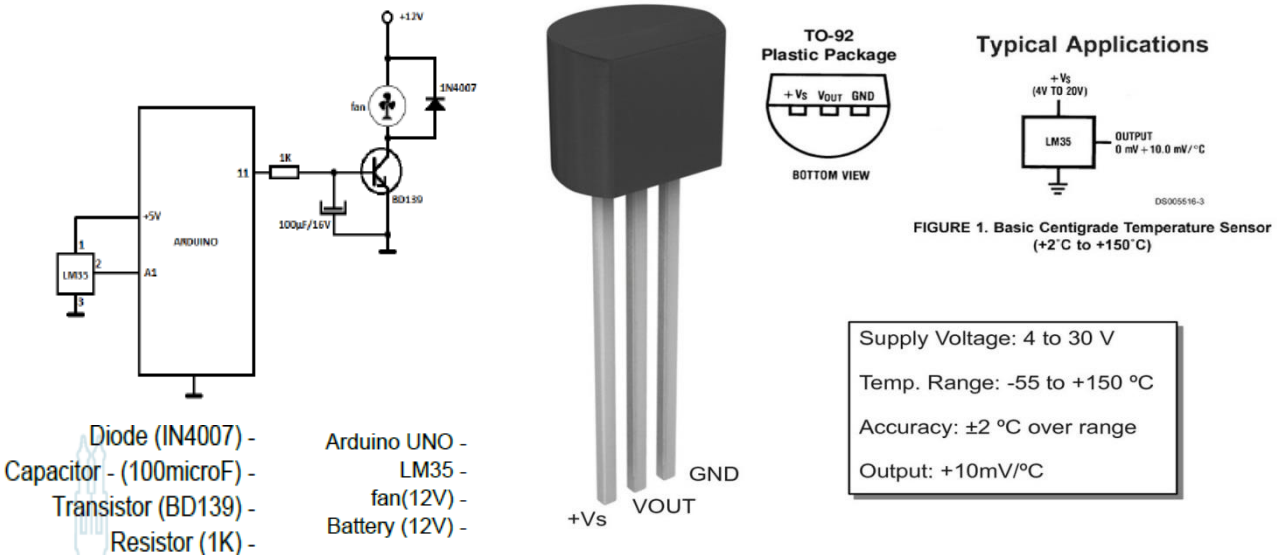
2- استخدام حساس الحرارة LM35 مع لوحة الأردوينو لتحكم سرعة مروحة .

تمهيد :

تعتمد فكرة التطبيق على التحكم في حرارة غرفة وذلك باستخدام مروحة تدور بسرعة أعلى كلما ارتفعت درجة حرارة الغرفة . وفي هذا المثال سوف نستبدل المروحة بآفوميتر لقياس الجهد المتغير ونلاحظ أنه كلما أعطي الحساس درجة حرارة أعلى ارتفعت قراءة الأفوميتر حتى يصل قراءة الأفوميتر الى 5 فولت عند درجة حرارة 60 درجة مئوية .

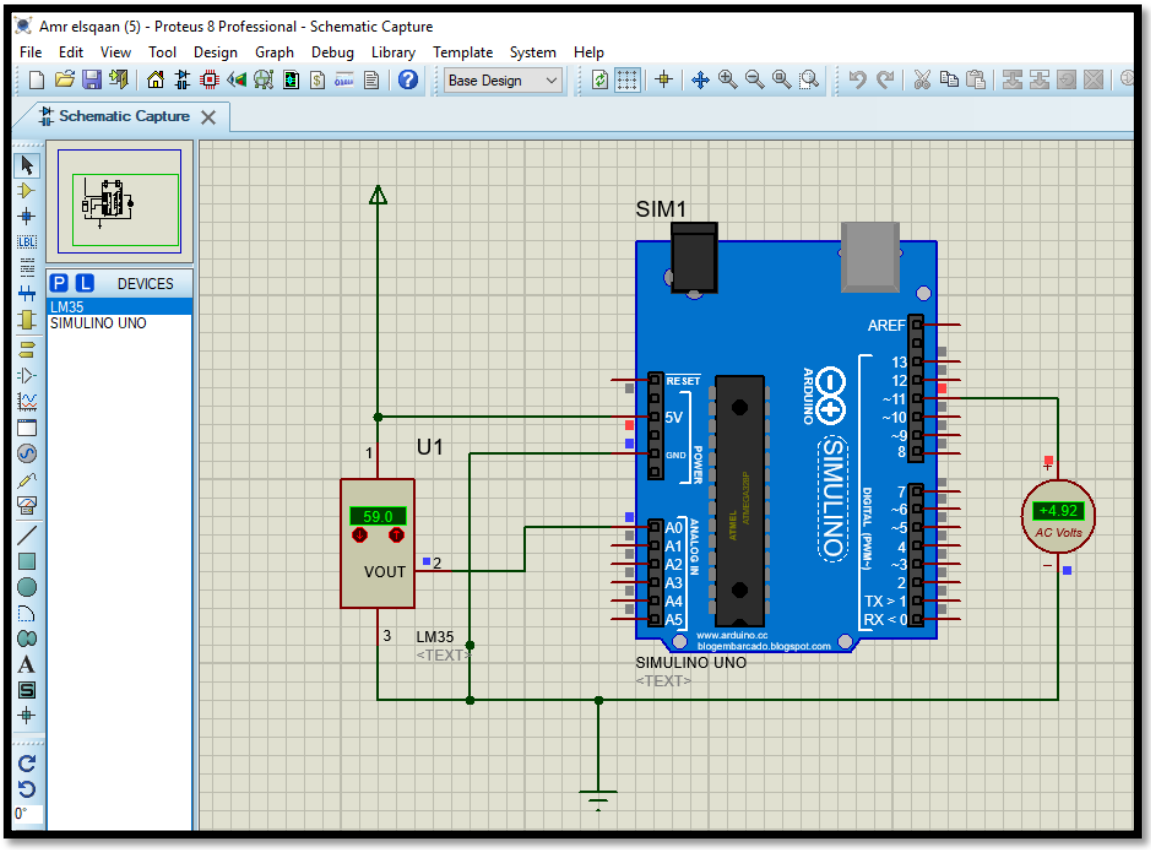
والجدير بالذكر أن هذا ال sensor عبارة عن ترانزستور حساسيته عالية لدرجة الحرارة , فكلما زادت درجة الحرارة كل ما زادت توصيليته (علاقة طردية) وعن طريق ال Datasheet الخاصة به نجد أن كل زيادة في درجة الحرارة بمقدار 1 °C يقابلها زيادة في فرق الجهد عليه بمقدار 100mV,

ولحساس الحرارة LM35 sensor ثلاث أطراف كما هو موضح بالشكل (25) مع ملاحظة إن الوجه المقابل لك هو الوجه المسطح وليس الدائري.



الشكل (25)

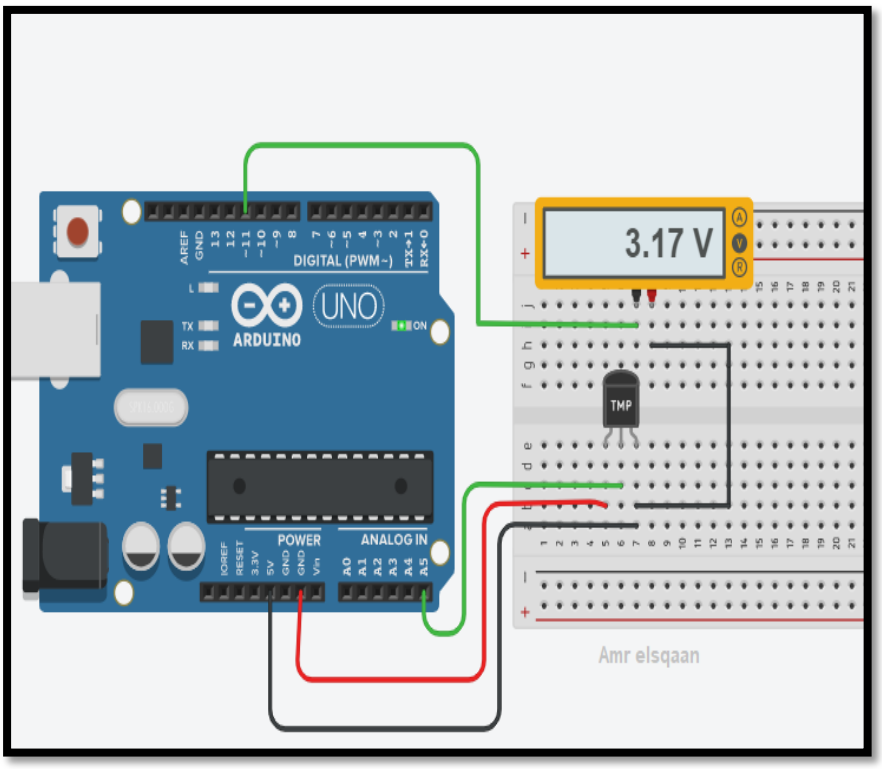
أ- المخطط باستخدام برنامج المحاكاة Proteus :



ب- كتابة الكود البرمجي لتشغيل الدائرة باستخدام برنامج Arduino IED:

```
int fan =11;
int tempSensor = A0;
int fanSpeed = 0;
float level,volt,temp;
void setup()
{
  pinMode(fan, OUTPUT);
}

void loop()
{
  level=analogRead(tempSensor);
  volt=level*(5.00/1024);
  temp=volt*100;
  if(temp>25 && temp<60)
  {
    fanSpeed = map(temp,25,60,0,255);
  }
  analogWrite (fan,fanSpeed);

}
```

ج – توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard



اسم التطبيـــــــق :	التحكم في درجات اللون لـ RGB LED	
زمن التطبيـــــــق :		رقم التطبيـق : 7
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- أسلاك توصيل	5- جهاز آفوميتر 6- EGB LED 7- عدد 3 مقاومات كربونية 100 أوم 8- عدد 3 مقاومات متغير 10 K

الهدف من التطبيق :

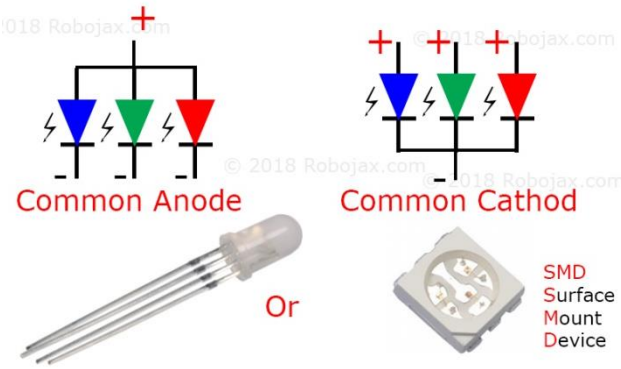
1- يهدف التطبيق الى التدريب على استخدام المداخل والمخارج التماثلية للوحة الأردوينو

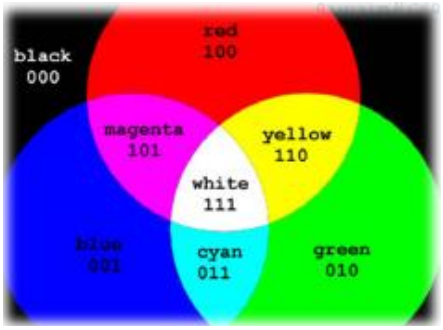
2- استخدام RGB LED مع لوحة الأردوينو للتحكم بلون الاضاءة .

3- توصيل المقاومات المتغيرة مع لوحة الأردوينو .

تمهيد :

RGB LED هو عبارة عن ثلاث موحدات ضوئية في عنصر واحد بحجم الموحد الضوئي العادي LED 5mm Diodes وله أربع أطراف لكل لون طرف والطرف الرابع لمصدر الجهد أو الأرضي حسب نوعه كما هو موضح بالشكل (26)





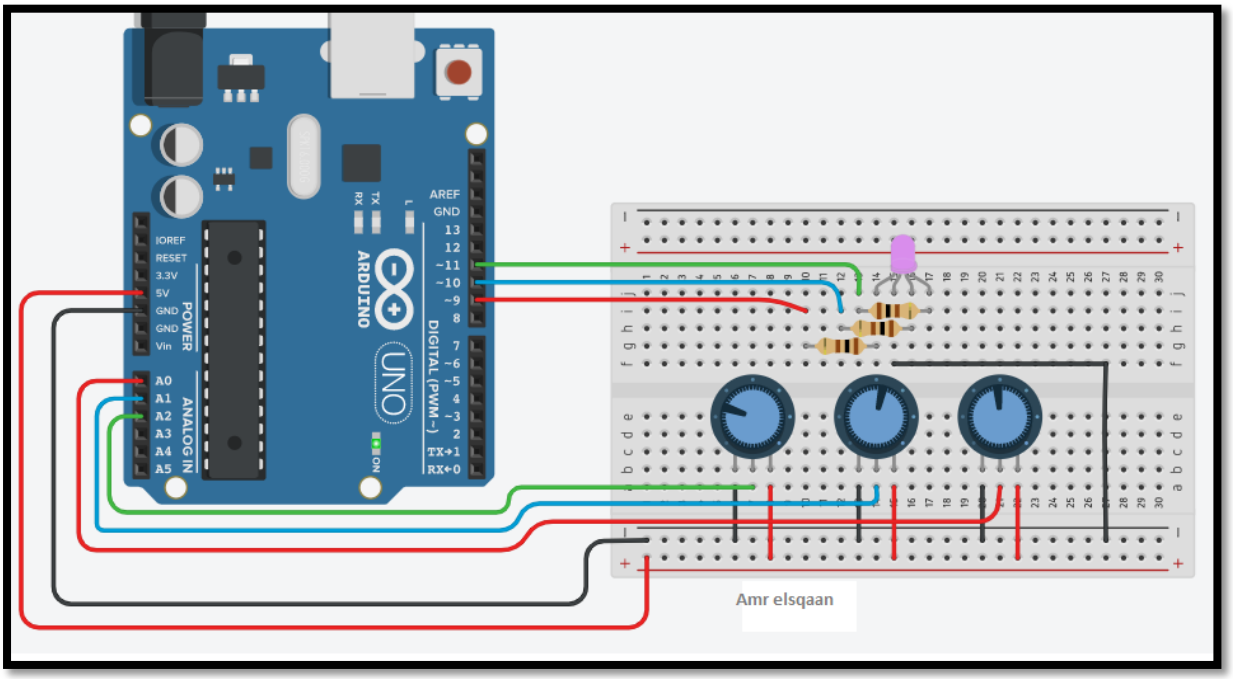
سنقوم باستخدام ثلاث مقاومات متغيرة للتحكم في شدة اضاءة كل لون في RGB على حدة باستخدام PWM وبالتالي نحصل على ألوان كثيرة ناتجة من اختلاط درجات الألوان الثلاثة الأساسية كما بالشكل المقابل .

الشكل(26) يوضح RGB LED

كتابة الكود البرمجي لتشغيل الدائرة باستخدام برنامج Arduino IED:

```
int red =9;
int green = 10 ;
int blue = 11 ;
int redPot = A0;
int greenPot =A1;
int bluePot = A2;
int redVal,greenVal,blueVl;
void setup(){
 pinMode(red,OUTPUT);
 pinMode(green,OUTPUT);
 pinMode(blue,OUTPUT);
}
void loop(){
 redVal = analogRead(redPot);
 redVal = map(redVal,0,1023,0,255);
 analogWrite (red,redVal);

 greenVal = analogRead(greenPot);
 greenVal = map(greenVal,0,1023,0,255);
 analogWrite (green,greenVal);

 blueVl = analogRead(bluePot);
 blueVl = map(blueVl,0,1023,0,255);
 analogWrite (blue,blueVl);
}

```

توصيل لوحة الأردوينو بعد برمجتها مع لوحة الاختبار breadboard :



عزيزي المتدرب :

- تقيد بالإرشادات والأنظمة المتبعة في معمل الحاسب الآلي.
- تقيد بلباس التدريب داخل الورشة والتزم بمتطلبات السلامة الأخرى
- لا تستخدم أي وسائط تخرين خارجية منعا لانتشار الفيروسات على جهاز الحاسب الآلي
- بعد الانتهاء من العمل على جهاز الحاسب تأكد من أنك أغلقت الجهاز بالشكل السليم
- لسلامتك تأكد من مناسبة جهد مصدر الطاقة المغذي لجهاز الحاسب قبل تشغيله
- احرص على الجلوس أمام الحاسب بطريقة سليمة محافظا على العمود الفقري

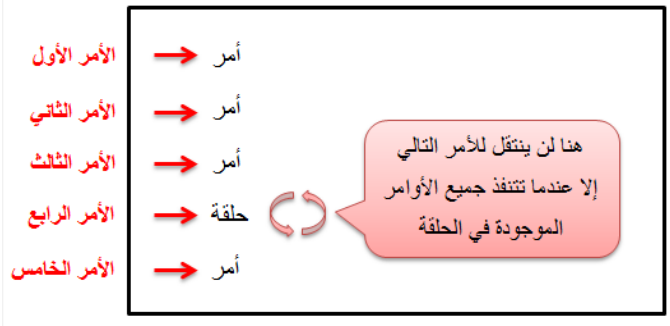
-

حلقات الأردوينو Arduino loops :

نستخدم الحلقات (Loops) بهدف عدم تكرار نفس الكود عدة مرات. إذاً أي كود نريد تنفيذه عدة مرات نقوم بكتابته داخل حلقة فتقوم هي بإعادة تنفيذ الكود قدر ما شئنا ضمن شروط معينة نقوم نحن بتحديدها.

طريقة تنفيذ الأوامر و الحلقات

الأوامر في العادة تنفذ بتسلسل وراء بعضها و لكن الحلقات تجعل سهم تنفيذ الأوامر يقف عندها فيقوم بتنفيذ الأوامر التي بداخلها عدة مرات, و بعد أن يخرج من الحلقة يعود و يكمل تنفيذ باقي الأوامر الموجودة بعدها كالتالي.



عند تنفيذ الحلقة فإن الأوامر الموضوعة فيها تنفذ بشكل منفصل عن باقي الأوامر الموجودة في البرنامج

أ- بنية (for Loop) :

```
for( initialisation; condition; increment أو decrement )
{
      //statements
}
```

حلقة for والتي تستخدم لتكرار جملة من التعليمات عدد من المرات يحددها المبرمج ، ولهذه الحلقة ثلاث بارامترات

initialisation: هي البارومتر الاول والتي يتم فيها وضع قيمة بدائية والتي سوف يبدأ منها العد ، وتعتبر الخطوة الاولي .

condition: وهو البارومتر الثاني وهو الشرط الذي سيتوقف عنده العد فما دام الشرط محقق فالحلقة سوف تستمر بالعمل ، وهذه ثاني خطوه

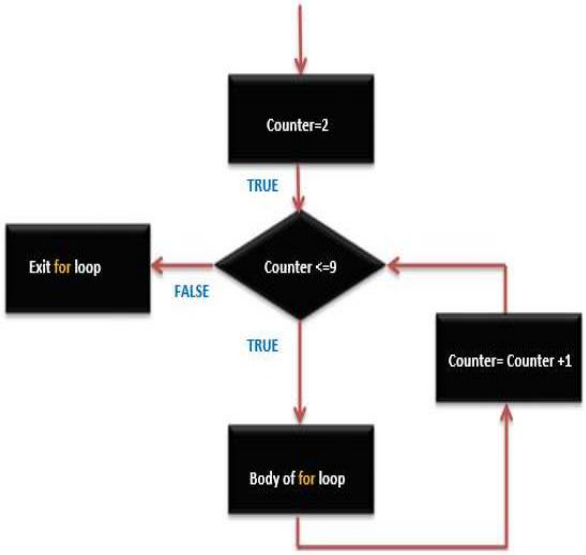
statements: هي الخطوة الثالثة, و تعني تنفيذ جميع الأوامر الموجودة في الحلقة و هي تتنفذ في كل دورة. بعد أن تتنفذ جميع الأوامر سيصعد إلى الخطوة الأخيرة التي تحدث في نهاية كل دورة و هي إما زيادة قيمة العداد أو إنقاصها.

decrement أو increment: هي البارامتر الثالث و هي تتنفذ في كل دورة. هنا نحدد كيف تزداد أو تنقص قيمة العداد, و لا نضع بعده .

تذكر فقط أن جميع هذه الخطوات تتكرر في كل دورة ما عدا أول خطوة و السبب أننا لا نحتاج إلى تعريف عداد جديد في كل دورة بل نستعمل العداد القديم و الذي من خلاله نعرف في أي دورة أصبحنا.

- مثال :

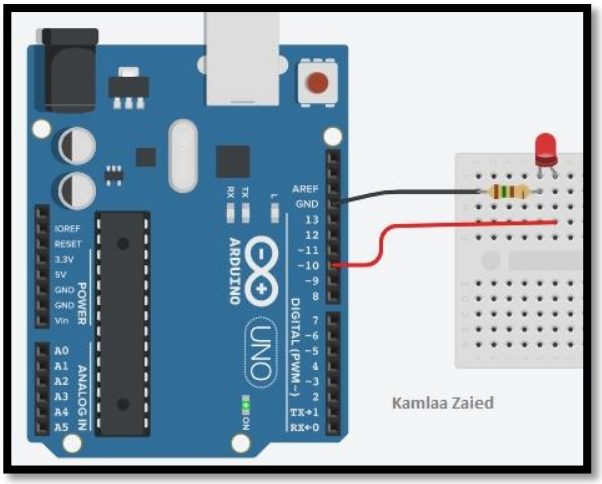
```
for(counter=2;counter <=9;counter++)
{
   //statements block will executed 10 times
   }
```



1- المتغير Counter وبه قيمة 2
2- الشرط هنا أن المتغير Counter أقل من أو يساوي 9 .
3- إذا ما تحقق الشرط يتم تنفيذ statements
4- مقدار الزيادة في كل دورة واحد

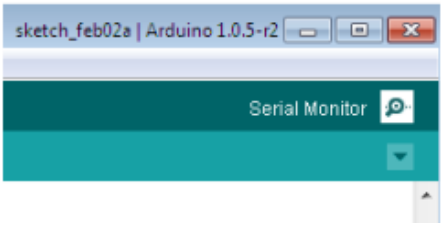
اسم التطبيـــــــق :	إضاءة ليد بالتدريج		
زمن التطبيـــــــق :		رقم التطبيـق :	8
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- لوحة Arduino Uno 4- أسلاك توصيل		5- LED 6- عدد 1 مقاومات كربونية 100 أوم

الهدف من التطبيق : يهدف التطبيق الى التدريب على استخدام جملة FOR .




```
// Dim an LED using a PWM pin
int PWMpin = 10;
void setup() {
  // no setup needed
}

void loop() {
  for (int i = 0; i <= 255; i++) {
    analogWrite(PWMpin, i);
    delay(10);
  }
}
```

نافذة الاتصال التسلسلي " شاشة السيريال "



هي أداة على الكمبيوتر (توجد في برنامج Arduino IDE) عبارة عن أيقونة أعلى اليمين تعمل علي التواصل مع الأردوينو أثناء تشغيله ، ويمكنك استخدامها لعرض المعلومات من الأردوينو إلى شاشة الكمبيوتر أو ارسال أرقام من الكمبيوتر الى الأردوينو أثناء التشغيل ، وفيما يلي :

1- الأوامر الأساسية لإظهار كتابة أو قيمة على شاشة السيريال :

- قبل استخدام شاشة السيريال يجب إعطاءها أمر البدء والعدد 9600 هو سرعة التراسل وهو مناسب للوحة الأردوينو أنو

```
Serial . begin (9600) ;
```

- أمر print لإظهار عبارة من الأردوينو إلى شاشة الكمبيوتر

```
Serial . print (" hey");
```

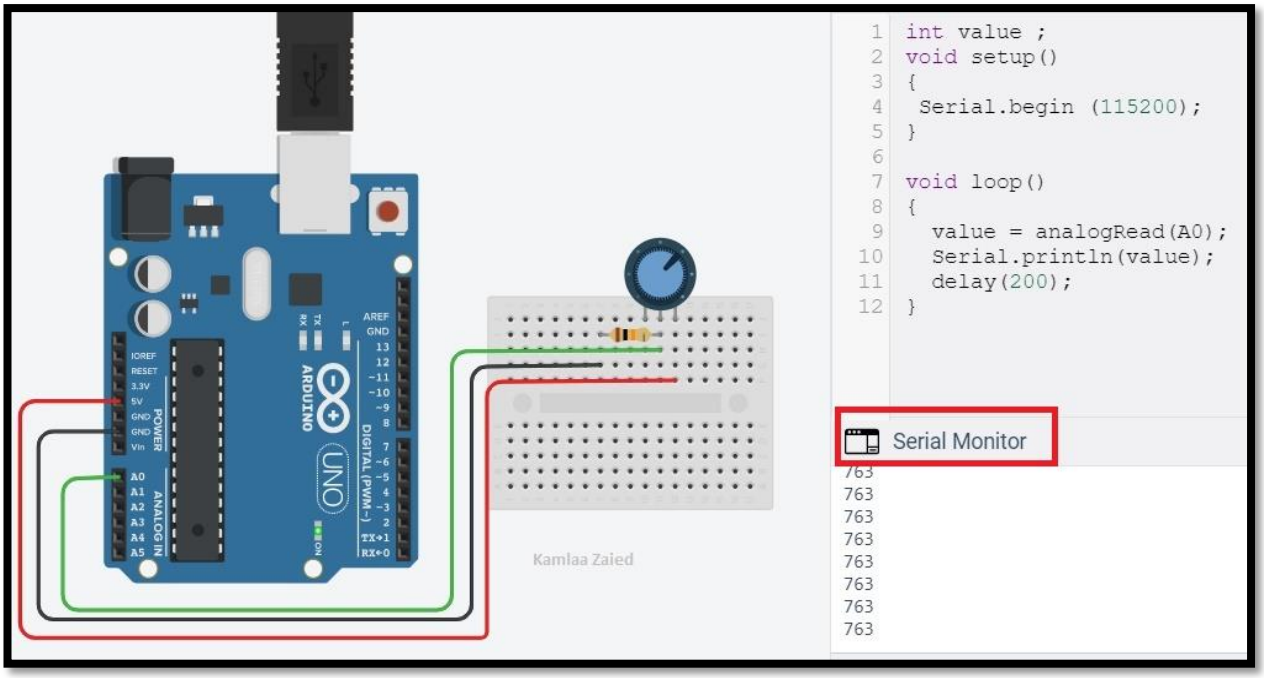
- \n لبدء سطر جديد و \t لترك مسافة tab

```
Serial . print (" \n\t");
```

- عند أضافة الحرفين (ln) بعد الأمر print فإنه يظهر العبارة ثم ينتقل الى السطر الجديد

```
Serial . println (x);
```

اسم التطبيـــــــق :	استخدام شاشة السيريال في قراءة قيمة مقاومة متغيرة		
زمن التطبيـــــــق :		رقم التطبيـق :	9
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard	3- مقاومة متغيرة 4- لوحة Arduino Uno	



2- الأوامر الأساسية لإرسال القيم من شاشة السيريال إلى الأردوينو أثناء تشغيله وتتم على ثلاث خطوات
1- إرسال رسالة تظهر على الشاشة وتشرح المطلوب من المستخدم
2- نوقف تنفيذ الكود بانتظار المستخدم لدخل قيمة
3- نستخدم الأمر المناسب لقراءة القيمة المرسلة من المستخدم

- يستخدم هذا الأمر للكشف إذا قام المستخدم بإرسال أي قيمة أثناء تشغيل الأردوينو عبر شاشة السيريال .
- استخدام الأمر whill توقف تنفيذ الكود بانتظار دخل
- استخدام الأمر if طريقة أخرى للكشف عن الدخل لكنها لا توقف الكود

```
Serial.availabe();
while(Serial.availabe()==0) { }
if (Serial.availabe()>0) { }
```

- قراءة عدد ووضعة في متغير من نوع int

```
Int x=Serial . parseInt();
```

- قراءة عدد ووضعة في متغير من نوع float

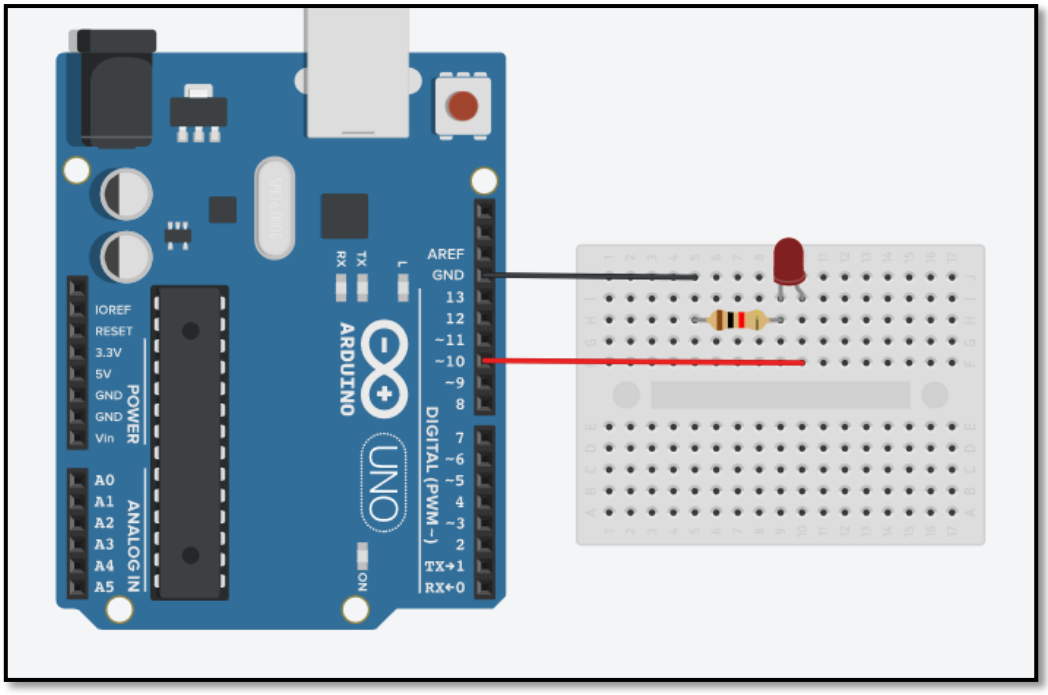
```
float x=Serial . parseFloat();
```

- قراءة نص ووضعة في متغير من نوع string

```
String x=Serial . readString ();
```

اسم التطبيـــــــق :	استخدام شاشة السيريال في سؤال المستخدم عن عدد مرات أضاء LED وسرعتها		
زمن التطبيـــــــق :		رقم التطبيـق :	10
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard	3- مقاومة 150أوم , LED 4- لوحة Arduino Uno	

أ- مخطط الدائرة



ب- الكود البرمجي والمدخلات من خلال شاشة السيريال

```
int LED =10;
void setup()
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println("how many blinks ?");
  while(Serial.available()==0){ }
  int n=Serial.parseInt();
  Serial.println("what is the delay in (ms)?");
  while (Serial.available()==0){ }
  int d= Serial .parseInt();
  for(int i=n;i>0;i--){
    digitalWrite(LED,HIGH);
    delay(d);
    digitalWrite(LED,LOW);
    delay(d);}
  }
```

Serial Monitor

```
how many blinks ?
what is the delay in (ms)?
```

Send Clear

ملاحظة : عند استخدام شاشة السيريال فإن لوحة الأردوينو تستخدم المنفذين الرقميين (0,1) للتواصل مع الحاسب لذا لا يمكنك استخدامهما كمنافذ في هذه الحالة

الدوال :

الدالة عبارة عن بلوك أو مجموعة من الاسطر البرمجية ذات علاقه بينهما . ويرجع السبب في استخدام الدوال إلى أنه في بعض الأحيان أثناء كتابة البرنامج نجد جزء من الكود ممكن أن نحتاجه في أكثر من مكان داخل البرنامج فبدلا من نسخ هذه الأكواد في هذه المقاطع من الكود يتم استخدام الدوال ، حيث يتم انشاء الدالة مرة واحدة وكتابة مجموعة من الاسطر البرمجية بداخلها وعند الرغبة في استخدام هذه الأكواد داخل البرنامج يتم ذلك فقط من خلال كتابة اسم الدالة فالدالة يتم انشائها مرة واحدة بينما يتم استدعائها أكثر من مرة .

مميزات استخدام الدوال في الكود البرمجي :

1- اختصار الكود البرمجي ، إذ يكتفي باستدعائه باسمة فقط لينفذ العمل المطلوب .
2- تلافي خطوات التكرار البرمجية نفسها.
3- اختصار عملية البرمجة وتنفيذ البرنامج بأسرع وقت ممكن .
4- توفر مساحة من الذاكرة المطلوبة للبرنامج .
5- تسهيل عملية مراجعة الكود وتصحيح الأخطاء والتعديل على الكود .

بنية الدالة في بيئة الأردوينو

```
Function _type  name ( Parameters )
{
Statement ;
Return result ;
}
```

1	Function _type	نوع القيمة التي سوف تعود بها الدالة علي سبيل المثال (int,float ,..) وفي حالة عدم إرجاع قيمة للدالة تكون الدالة من النوع void
2	name	اسم الدالة
3	Parameters	عبارة عن متغير من أي نوع مثل (int أو double أو String..) يتم تعريفه بين أقواس الدالة. و من خلاله يجعل المستخدم يدخل قيمه الخاصة في الدالة فتتغير النتيجة التي ترجعها الدالة حسب القيم التي يقوم هو بإدخالها ، ولا توضع لها قيم في حالة عدم الحاجة لها .
4	Statement	البنية البرمجية للدالة
5	Return result	القيمة التي سوف يتم اعادتها ، وإذا لم يكن هناك قيمة معادة فلا حاجة لهذه التعليمة

أنواع الدوال :

1- دالة ليس لها دخل وليس لها خرج .
2- دالة لها دخل وليس لها خرج .
3- دالة ليس لها دخل ولها خرج .
4- دلة لها دخل ولها خرج .

أمثلة لكود و دالة تعمل على تحويل درجة حرارة من فهرنهايت الى منوية: مع العلم أن :

$$C = (F - 32) * \frac{5}{9}$$

```
int F1=0; //here put the fahrenheit temp
int F2=10;
int F3=25;

void setup(){
int  C1=FtoC(F1);  //FtoC will convert Fahrenheit to degrees
int  C2=FtoC(F2);
int  C3=FtoC(F3); }

void loop(){ }

int FtoC(int x){  //FtoC is a function to convert Fahrenheit to degrees
  int C = x-32*5/9;
  return(C);}
```

مثال لعمل دالة (Blink) ترسل لها قيمتين عدد مرات الوميض و التأخير الزمني.

```
int LED = 13;
void setup(){
  Blink(10,300);
  Blink(5,1000);
  Blink(2,5000);}

void Blink(int T, int D){
  for (   ; T>0 ; T--){
    digitalWrite(LED,HIGH);
    delay(D)
    digitalWrite(LED, LOW);
    delay(D); } }
```

المصفوفات :

عند التعامل مع حجم كبير من البيانات كنا نستخدم الكثير من المتغيرات لتخزين ومعالجة تلك البيانات وهذا يجعل البرنامج طويل ومعقد لكن باستخدام المصفوفة يستطيع المبرمج استعمال المتغيرات بشكل أقل وفي بعض المكتبات يجب عليك ارسال البيانات في شكل مصفوفات .

مثال: نود جعل المنافذ 0،1،3،4،6،7،9،10،12،13 مخارج OUTPUTS
بينما المنافذ 2،5،8،11 نريدها أن تكون منافذ دخل مع ربطها بمقاومة رفع الجهد الداخلية PULLUP
بدل أن نكتب الأوامر في 14 سطر ، بإمكاننا الاستفادة من المصفوفات كما يلي:

```
int OT[]={0,1,3,4,6,7,9,10,12,13};
int IN[]={2,5,8,11};
for(int i=0; i<10; i++){ pinMode(OT[i] ,OUTPUT);}
for( i=0; i<4;i++){ pinMode(IN[i],INPUT);}
```

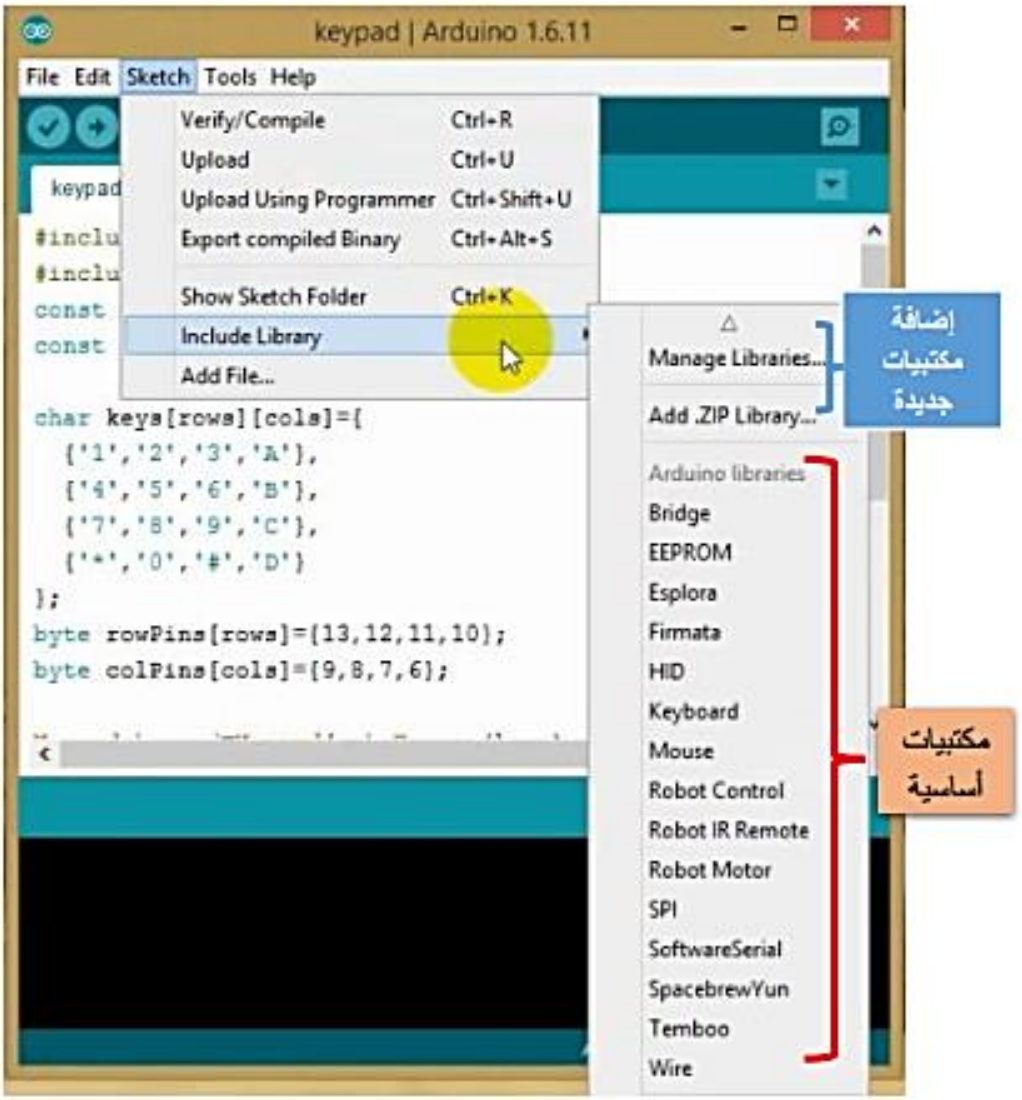
المكتبات :

هي مجموعة من الإجراءات البرمجية المستقلة التي يمكن استدعاؤها من قبل البرامج لتنفيذ وظائف تكمل عملها ... . فالمكتبات ليست إلا عدد من الدوال تمت إضافتها معا في ما يسمى بالمكتبة لتسهيل الوصول لها حال الحاجة اليها فاستخدام المكتبات يوفر علينا الكثير من الجهد والوقت لأن فكـرة البرمجـة من الصفـر، فكرة غيـر مجديـة في أغلـب الأحيـان. لمـاذا؟ لأنك تعيـد بنـاء شيء موجود مسبقـاً يمكنك استخدامه والاستفادة منه كيفمـا تريـد. فهذه المكتبات مخصصة للتعامل مع الموديلات المختلفة كموديول الإرسال والاستقبال للموجات الراديوية RF وغيرها أو تكون مخصصة للطرفيات كالشاشات بأنواعها أو المحركات أو الحساسات المتنوعة ,

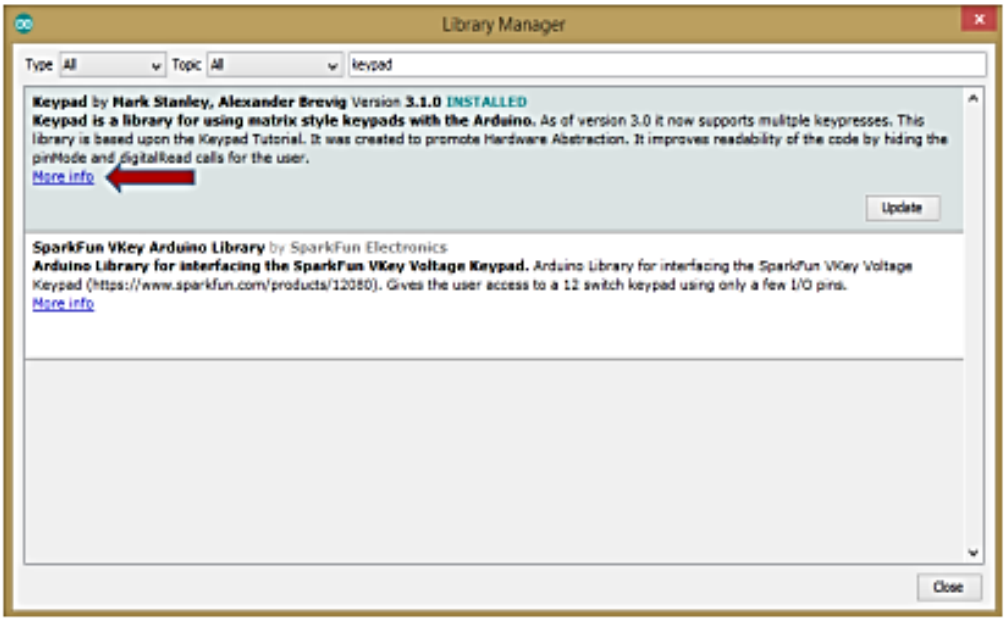
أضافة المكتبة الى برنامج Arduino IDE

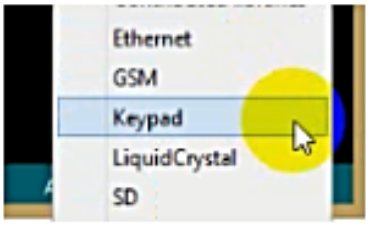
١- من القائمة sketch نختار إضافة مكتبة Include Library.

٢- تظهر قائمة تضم العديد من المكتبيات الموجودة أساسا في البرنامج، نختار منها المكتبة المطلوبة.



٣- أما إذا كانت المكتبة المطلوبة غير موجودة فممكن إضافتها عبر خيار إدارة المكتبيات Manage Library فتظهر لنا النافذة التالية:





يتم من خلال هذه النافذة البحث عن المكتبة المطلوبة (مثلا في النافذة السابقة يتم البحث عن المكتبة الخاصة بلوحة المفاتيح الست عشرية keypad) وإضافتها لقائمة المكتبيات الموجودة أساسا في البرنامج، فيتم الضغط عليها لإضافتها للكود البرمجي، ولعرض المزيد من المعلومات حول المكتبة المطلوبة نختار الأمر More info فيتم عرض صفحة تحتوي على شرح عن المكتبة ومثال عن استخدامها.

٤- يتم إضافة المكتبيات أيضا التي تكون على شكل ملفات مضغوطة " zip " عبر الخيار Add ZIP Library فيتم استعراض الجهاز لتحديد مكان المكتبة ومن ثم إضافتها.

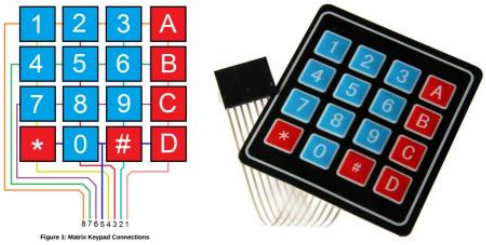
عزيزي المتدرب:

تركيزك في التدريب يزيد من قدراتك المهارية ويقيك من المخاطر .

الطرفيات :

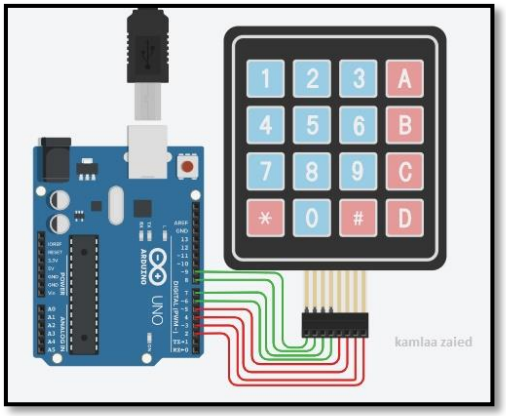
➢ لوحة المفاتيح :

الشكل (27) يوضح طريقة توصيل لوحة المفاتيح (4x4) على لوحة الأردوينو، ولكي يمكننا التعامل مع لوحة المفاتيح لابد من استخدام المكتبة الخاصة بها وأن كان هناك الكثير من المكتبات التي تستخدم في تشغيل لوحة المفاتيح (4x4) ولكن سوف نعرض أحد هذه المكتبات .



الشكل (27) يوضح لوحة المفاتيح 4x4

➢ الكود البرمجي للضغط على أي مفتاح من لوحة المفاتيح فيظهر في شاشة السيريال .




```
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the key
byte colPins[COLS] = {5, 4, 3,2}; //connect to the column pinouts of the k

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();

  if (key){
    Serial.println(key);
  }
}
```

تمرين (3)

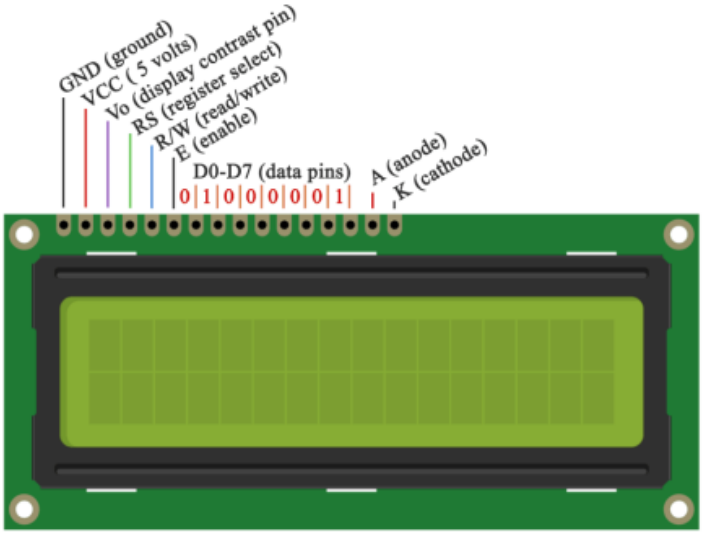
باستخدام برنامج المحاكاة ولوحة الاردوينو :

استخدم لوحة المفاتيح keypad بحيث يضئ LED بالضغط على المفتاح (*) لا يعمل LED بالضغط على المفتاح (#)

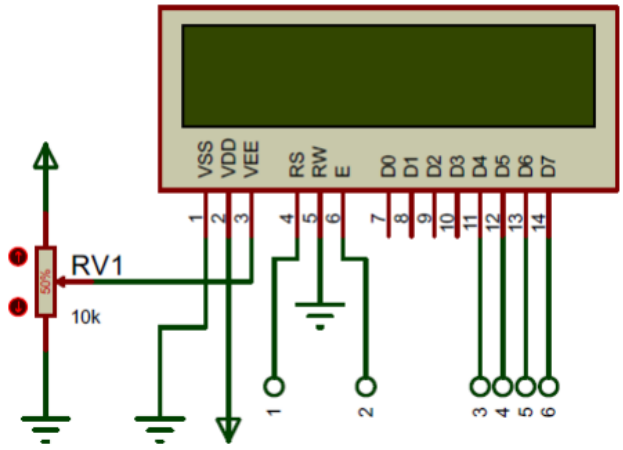
(الحل في دليل حل التمارين بنهاية الوحدة)

➢ الشاشة LCD :

شاشة العرض البلوري السائل أو بالاختصار LCD Liquid Cristal Display، تعتبر شاشة العرض الكرستالية من أهم وسائل عرض البيانات المرسلة إليها من لوحة الآردوينو وتتميز بسهولة التعامل وانخفاض التكلفة المادية وإمكانية عرض البيانات وبعض الرموز والأشكال، كما تحوي بداخلها معالج خاص لتسهيل التعامل معها حيث يكون هو المسؤول عن العمليات المعقدة للإظهار على الشاشة كما تحتوي على ذاكرة لحفظ المحارف المطلوب عرضها على الشاشة , تتوفر الشاشة بعدة قياسات و أحجام و تقاس عادة هذه الشاشة بعدد الأسطر و الأعمدة التي تحتويها الشاشة، لكن و مهما اختلفت أبعاد الشاشة يبقى لها نفس الأقطاب الخارجية للتحكم :



- أقطاب البيانات D0 ~ D7.
- قطب التغذية VDD = 5v.
- قطب الأرضي Vss = GND.
- قطب التحكم بتباين الشاشة Vee .
- قطب التفعيل E.
- قطب الضبط RS.
- A & K موجب وسالب ليد إضاءة الشاشة.



توصيل شاشة العرض LCD مع الأردوينو

تشغيل شاشة LCD مع الأردوينو :

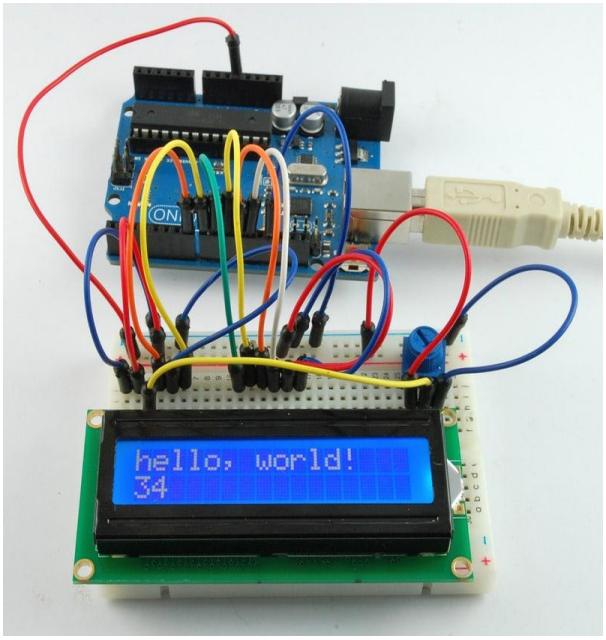
تضمن بيئة التطوير Arduino IDE مكتبية خاصة بتشغيل شاشة العرض الكرستالية LCD نجدها في قسم المكتبيات بالاسم LiquidCrystal ولهذه المكتبية عدة بارامترات يجب ضبطها قبل الشروع في العمل على الشاشة كما توفر هذه المكتبية تعليمات العرض المختلفة ومسح البيانات والتحكم بموقع المؤشر على الشاشة وإمكانية إنشاء محرف جديد كل هذا سنجده من خلال أدوات هذه المكتبية.

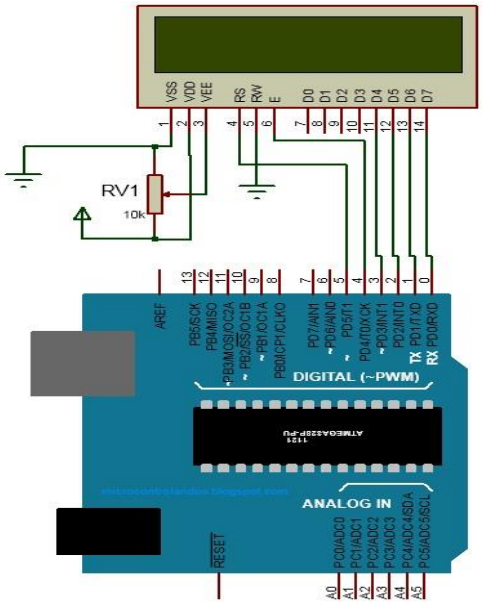
التعليمة	شرح التعليمة
`#include <LiquidCrystal.h>`	تضمين مكتبية تشغيل شاشة العرض الكرستالية للكود.
`LiquidCrystal lcd(RS,E,D4,D5,D6,D7);`	تحديد الأقطاب التي تم توصيل الشاشة إليها على لوحة الآردوينو مع المحافظة على هذا الترتيب للأقطاب.
`lcd.bgin(cols,rows);`	تحديد أبعاد الشاشة في قسم التهيئة، وذلك بتحديد عدد الأعمدة cols والأسطر rows.

`lcd.print( );`	طباعة البيانات على الشاشة، متحولات وعبارات.
`lcd.write( );`	طباعة العبارات فقط على الشاشة.

lcd.setCursor(col,row);	وضع مؤشر الكتابة في الموضع المحدد بالسطر والعمود على اعتبار أن السطر الأول والعامود الأول يحملان الرقم 0.
lcd.clear( );	مسح الشاشة.
lcd.blink( );	جعل المؤشر في حالة خفقان.
lcd.noBlink( );	إيقاف خفقان مؤشر الكتابة.
lcd.cursor( );	تشغيل المؤشر.
lcd.display( );	تشغيل الشاشة.
lcd.noDisplay( );	إطفاء الشاشة.
lcd.rightToLeft( );	تحديد جهة الكتابة من اليمين إلى اليسار.
lcd.leftToRight( );	تحديد جهة الكتابة من اليسار إلى اليمين.
lcd.autoscroll( );	تمرير النص خانة نحو اليمين.
lcd.write(Serial.read());	طباعة البيانات التي تصل من النافذة التسلسلية على شاشة العرض الكرستالية.

اسم التطبيـــــــق :	استخدام شاشة LCD وإظهار جملة Hello World		
زمن التطبيـــــــق :		رقم التطبيـق :	11
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard	3- شاشة LCD – مقاومة متغيرة 10K 4- لوحة Arduino Uno	





الكود البرمجي :

HelloWorld | Arduino 1.8.2

File Edit Sketch Tools Help

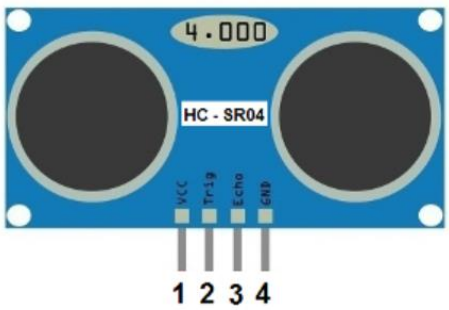
HelloWorld §

```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

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

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);
}
```

➢ حساس الأمواج فوق الصوتية Ultrasonic Sensor

في هذا البحث سوف نستخدم حساس قياس المسافات بالأمواج فوق الصوتية والذي يأخذ الرمز HC-SR04، يعتمد هذا النوع من الحساسات على ارسال أمواج فوق صوتية عبر الهواء وعند اصطدام هذه الأمواج بأي جسم أو عائق تنعكس وتعود للحساس ومن خلال حساب الزمن الذي تستغرقه الموجة في عملية الذهاب والإياب يتم معرفة المسافة التي يبعدها الجسم عن الحساس وذلك على اعتبار أن سرعة الأمواج الصوتية في الهواء ثابتة.

أطراف الحساس الأمواج فوق الصوتية Ultrasonic Sensor



Pin 1: VCC
Pin 2: Trig
Pin 3: Echo
Pin 4: GND

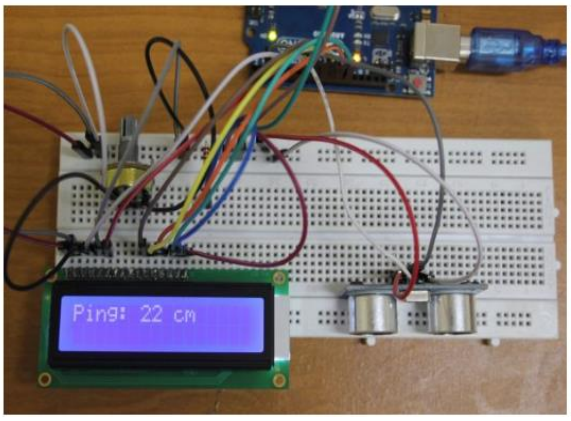
- قطب التغذية الموجب Vcc
- قطب التغذية السالب GND
- قطب القدح Trigger Pin الذي يعطي أمر للحساس بإرسال الأمواج فوق الصوتية
- قطب استقبال الموجة المنعكسة Echo Pin

اسم التطبيـــــــق :	استخدام حساس الموجات قوق الصوتية لقياس المسافة التي يبعدها جسم ما وطباعة المسافة على شاشة LCD		
زمن التطبيـــــــق :		رقم التطبيـق :	12
الادوات المستخدمة :	1- جهاز كمبيوتر 2- لوحة اختبار Breadboard 3- مقاومة متغيرة 10كيلو اوم	4- حساس HC-SR04 / شاشة LCD 5- لوحة Arduino Uno 6- مقاومة كربونية 220 أوم	

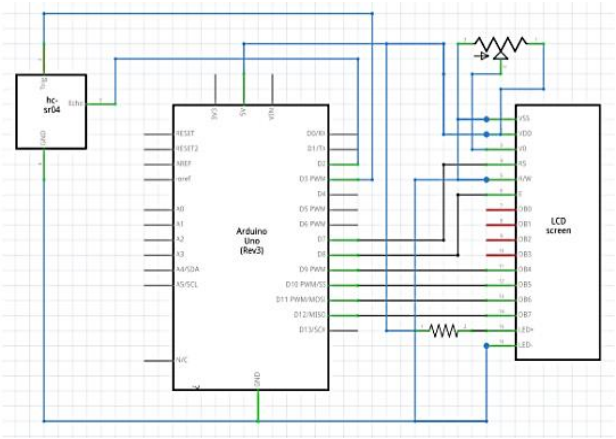
أ- تمهيد :

في هذا المشروع سنتعلم فكرة عمل حساس الموجات فوق الصوتية و كيفية استعماله لحساب المسافة بين الأردوينو و كائن ما. قد يستخدم في أنظمه الإنذار او في روبوت تخطي العقبات ثم سنقوم بعرض هذه القراءات على شاشة LCD

ب- مخطط الدائرة :



الدائرة العملية



المخطط النظري

جـ - الكود البرمجي :

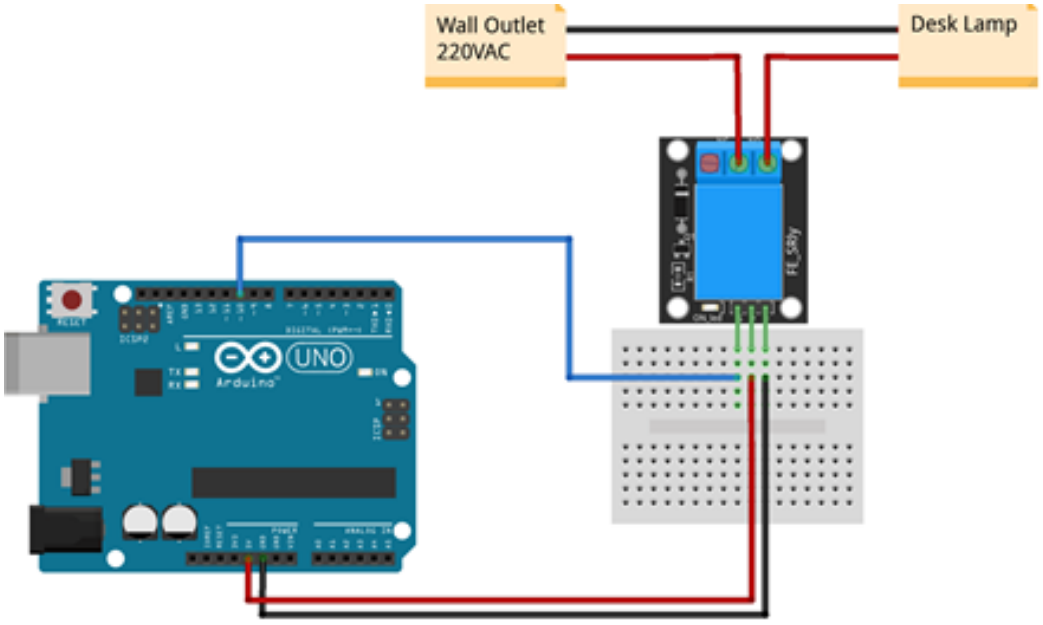
```
#include <LiquidCrystal.h>          // the LCD Library
#include <NewPing.h>                // the Ultrasonic Library

#define TRIG  3                     // Pin-3 of Arduino Connected to Trig Pin of Ultrasonic
#define ECHO  2                     // Pin-4 of Arduino Connected to Echo Pin of Ultrasonic
#define MAX_DISTANCE 100            // Max. Distance the Sensor Can Measure, Required for the Library


NewPing sonar(TRIG, ECHO, MAX_DISTANCE); // Ultrasonic Variable
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);  // LCD Variable


unsigned int duration;              // Variable Used To Store The Measured Distance

void setup()
{
 pinMode(TRIG,OUTPUT);              // Set Trig Pin of Arduino As Output
 pinMode(ECHO, INPUT);              // Set Echo Pin of Arduino As Input
 lcd.begin(16,2);                   // To Setup and Start the LCD
 lcd.clear();                       // To Clear the LCD
}

void loop()
{
 duration = sonar.ping();           // Send ping, get ping time in microseconds (uS).
 duration = duration / 2 / 29.4;    // Explained Below
 lcd.print("Ping: ");               // to Print " Ping:  " on the LCD
 lcd.print(duration);               // next to it print the Calculated Value
 lcd.print(" cm");

 // on lcd we will see "  Ping: 30 cm  "

 delay(1000);                       // min. delay between pings is 29 msec
 lcd.clear();                       // to clear the LCD for the Next Round
}
```

ملاحظة : هناك الكثير جدا من الطرفيات أو الحساسات التي تعمل مع الأردوينو والذي لا يسع وقت الوحدة لتناولها ولكن تستطيع عزيزي المتدرب من خلال شبكة الأنترنت الاطلاع عليها ومعرفة طريقة تشغيلها مع الأردوينو

المخرج التعليمي (3)	ينفذ منظومة تحكم مادية " Hard Wear " تعمل بالأردوينو		
معيـــــار الأداء (1)	ينفذ كارت إلكتروني Relay Module يعمل مع الأردوينو وفقا لمتطلبات التشغيل		
زمن التنفيـــــــــذ		أدلة التعليم	مرفق 6
معلومات إثرائيـــــه	فيديو يشرح فيه طريقة عمل Relay Module		

وظيفة دائرة Relay Module

- لا تستطيع لوحة الأردوينو إمداد تيار كافي لتشغيل الكثير من الدوائر الإلكترونية الملحقة بيها ، لذا جاءت الحاجة الى استخدام دائرة ربط Relay Module كدائرة وسيطه بين لوحة الأردوينو كدائرة تحكم وبين الدائرة الإلكترونية كدائرة متحكم فيها ، حتى تعمل الدائرة الإلكترونية دون تحميل لوحة الأردوينو عبئ سحب تيار عالي منها مما يؤدي إلى تلفها
- تأتي دائرة Relay Module بأشكال متعددة حسب عدد الدوائر الإلكترونية المراد ربطها مع لوحة الاردوينو ، فيوجد منها Relay Module بقناه واحدة أي تستخدم لربط دائرة إلكترونية واحدة مع لوحة الأردوينو أو بقناتين أو أربع قنوات أو ثمان قنوات والشكل (27) يوضح توصيل Relay Module مع لوحة الأردوينو .

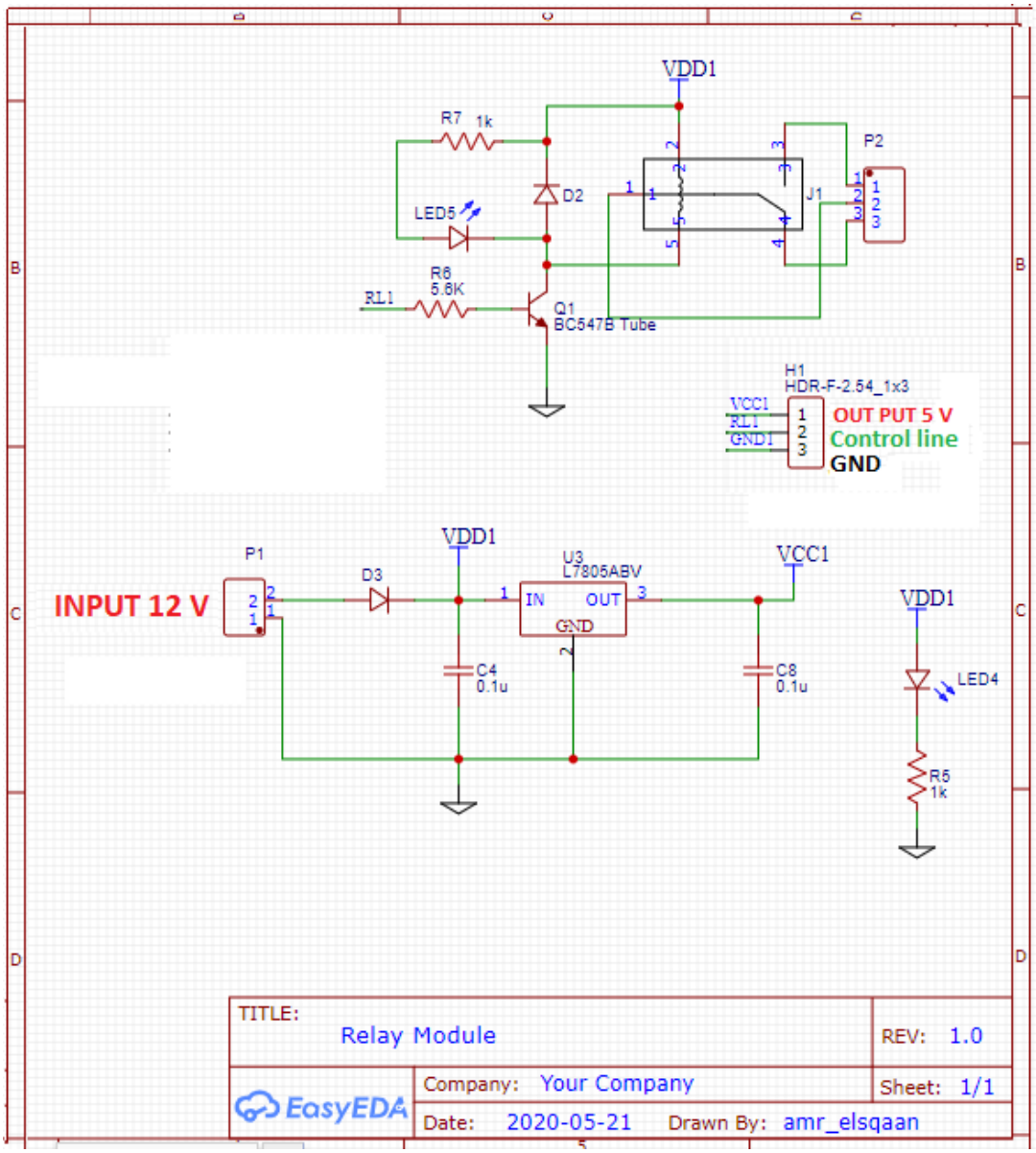


الشكل (27) يوضح طريقة توصيل Relay Module مع لوحة الأردوينو لتحكم في انارة لمبة تعمل على 220 فولت

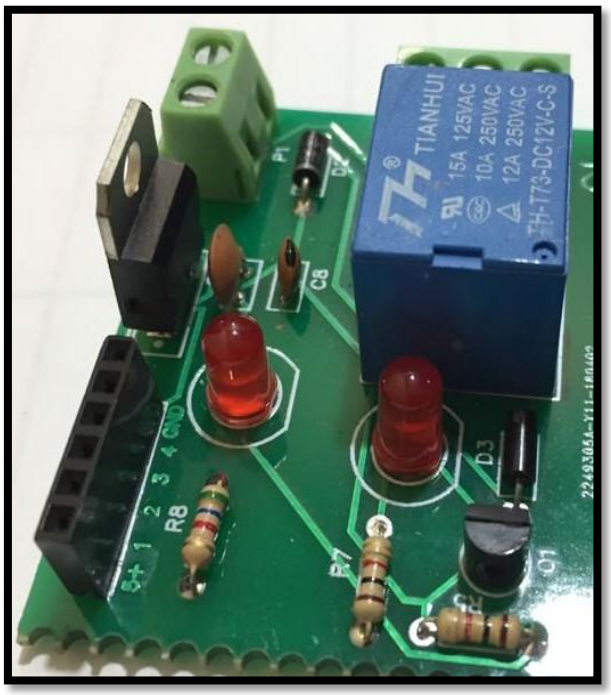
اسم التطبيـــــــق :	تنفيذ لوحة Relay Module بقناه واحدة	
زمن التطبيـــــــق :		رقم التطبيـق : 13
العناصر الالكترونية المستخدمة :	- Header 1x3, 1x2 , 1x3 - C 0.1 UF X 2 - D 1N4001 X2 - LED X2	- IC L7805 - Q BC547B - Relay 12VDC - R 5.6 K

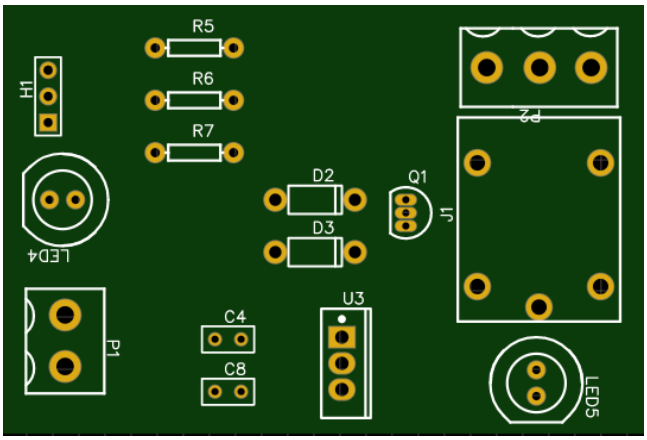
أ- المخطط النظري:

مضاف للوحة Relay Module دائرة متكاملة L7805 للحصول على جهد ثابت 5V لاستخدامه في تغذية أي دوائر إلكترونية بدلا عن استخدام لوحة الأردو ينو في الحصول على 5V و يتم استخدام أحد برامج تصميم الدوائر الإلكترونية التي تم دراستها في رسم وتحويل الدائرة النظرية الى لوحة PCB

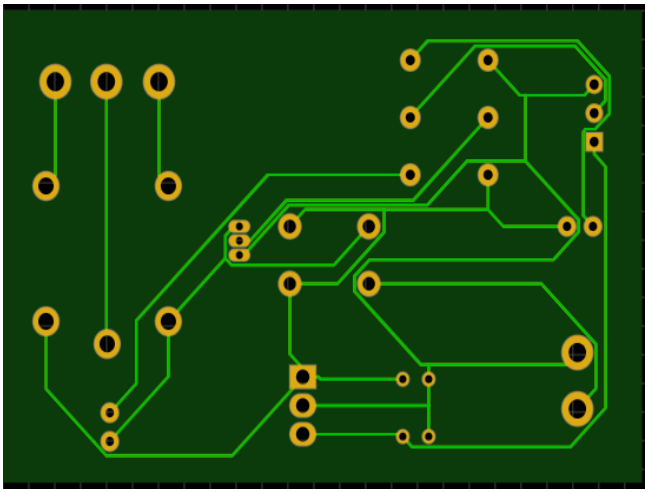


ب- لوحة PCB بعد تصميمها باستخدام أحد برامج تصميم الدوائر الإلكترونية :





TOP Side



Bottom Side

تمرين (3)

باستخدام دائرة Relay Module التي قمت بتنفيذها :

1- قم ببرمجة لوحة الأردوينو بكود برمجي يضيئ لمبة 220 V ثم ينتظر ثانية واحدة ثم يطفأها وهكذا لعدة مرات .

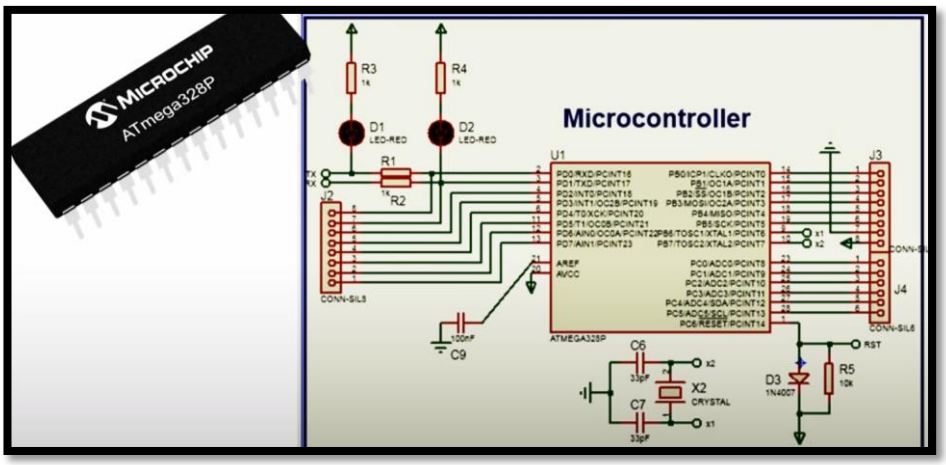
عزيزي المتدرب :

- احذر أثناء عملية توصيل الدائرة لأنك سوف تتعامل مع جهد 220 فولت .
- افصل القدرة الكهربائية (Turn off) عن الدائرة عند تركيب أو فصل عنصر من الدائرة

المخرج التعليمي (3)	ينفذ منظومة تحكم مادية " Hard Wear " تعمل بالأردوينو		
معيـــــار الأداء (2)	يصمم لوحة أردوينو وفقا لمتطلبات التشغيل		
زمن التنفيـــــــــذ		أدلة التعليم	مرفق 7
معلومات إثرائيـــــه	مقطع فيديو يوضح طريقة صنع الأردوينو الخاص بك		

في كثير من الأحيان نحتاج الى تصنيع لوحة أردوينو بمواصفات خاصة تختلف عن الموجودة في السوق من حيث الحجم والمخارج والمداخل لاستخدامها في مشروع تحكم يستمر لوقت طويل ، لذا يجب أن يكون لدينا القدرة على أنتاج لوحة أردوينو خاصه بنا نستطيع استخدامها في المشاريع المختلفة بما تتناسب مع طبيعة المشروع . وتتكون لوحة الأردوينو من ثلاث أجزاء رئيسية :

الجزء الأول : دائرة الميكروكنترولر Microcontroller

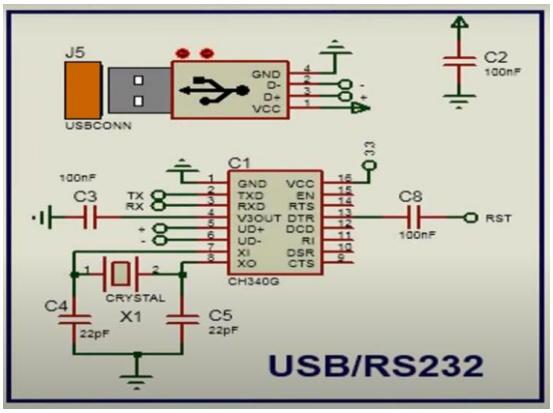


ويتكون من المتحكمة ATmega328p وبلورة كريستال 16MHZ وتعتبر قلب الميكروكنترولر وبدونه لا يعمل ويتصل معها مكثفين بسعة 33p لتقليل من الضوضاء وزيادة الثبات وأحيانا تأتي الكرستالة ومتصل معها المكثفين داخليا داخل مغلف واحد ، كذلك نصل المقاومة R5 لتفادي حدوث RST في كل مرة يعمل فيها الميكروكنترولر إلا إذا أردنا ذلك فيكون من خلال الضغط على مفتاح RST ، ثم نضيف المخارج والمداخل الرقمية والتناظرية للأرجل المتبقية من الميكروكنترولر .

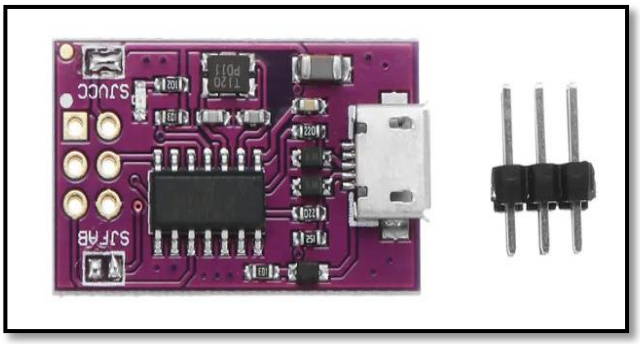
الجزء الثاني : الجزء الخاص بتبادل البيانات بين الميكروكنترولر والحاسب الالي:

من المعلوم أن ATmega328p لا يتعامل مع البيانات ألا عن طريق برتوكول التخاطب التسلسلي RS232(TX/RX) بينما الحاسب الألي يتعامل مع البيانات عن طريق برتوكول USB لذا لا بد من توافر دائرة تقوم بعمل المترجم بين البروتكولين لنقل البيانات من والى لوحة الأردوينو ، ولكن في حالة تصميم لوحة الأردوينو الخاص بنا والتي

نستخدمها في بناء مشروع تحكم يستمر لفترات زمنية كبير لا نحتاج فيها لوحة الأردوينو للاتصال بالكمبيوتر إلا لمرة واحدة فقط لتحميل البرنامج الذي تم تصميمة ، لذا نستطيع ان نستغنى عن هذه الدائرة ونوفر من التكلفة وحجم اللوحة ونستغنى عن هذه الدائرة بموديول خارجي Programmer Bootloader يستخدم لنفس الغرض .



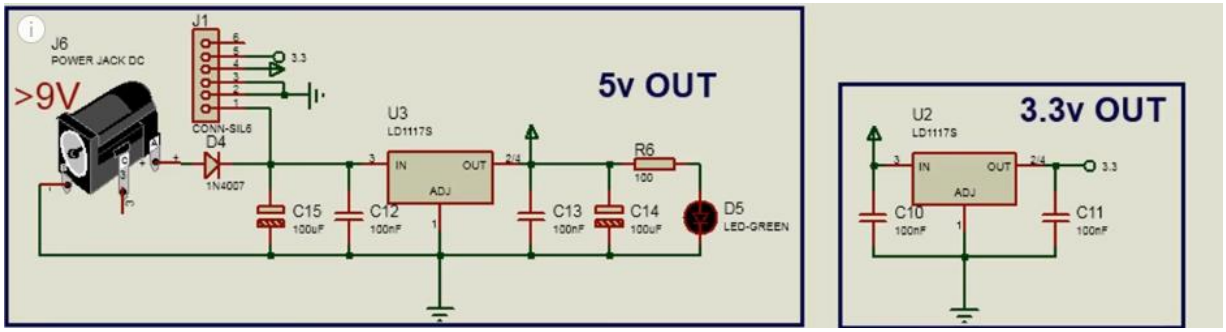
دائرة توضح الجزء الثاني من لوحة الأردوينو



دائرة Programmer Bootloader خارجية

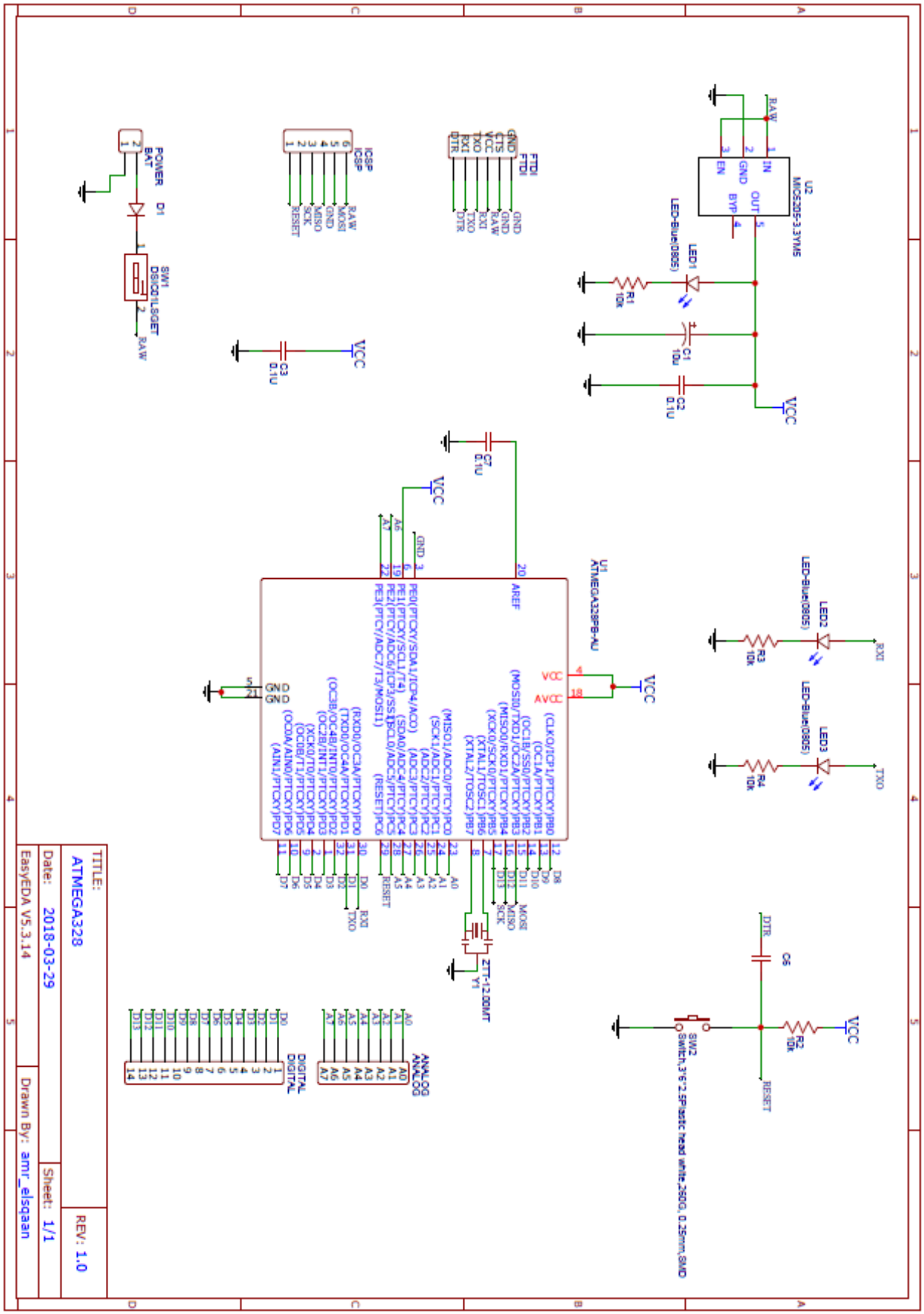
الجزء الثالث : دائرة التغذية :

ويتكون من دائرتين لتنظيم الجهد ، الأولي منظم جهد من 9V تقريبا إلى 5V أما دائرة المنظم الثاني في تنظم الجهد من 5V الى 3.3 V نظرا إلى أن أغلب الحساسات تعمل عند هذا الجهد والشكل التالي يوضح دائرتين تنظيم الجهد للوحة الأردوينو .

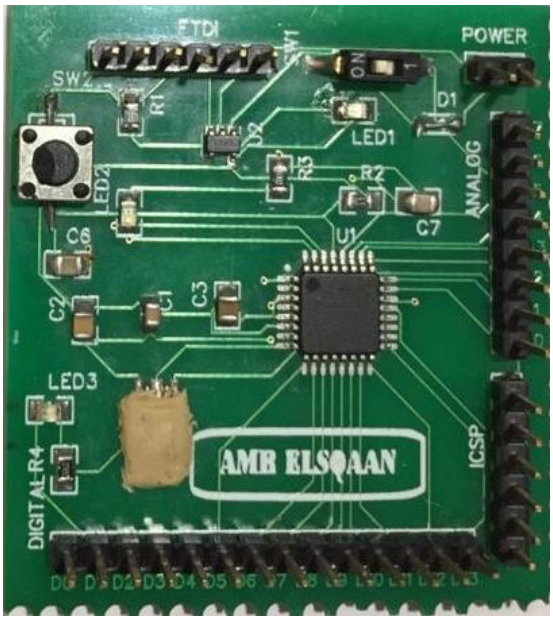


اسم التطبيـــــــق :	تصميم لوحة أردوينو باستخدام أحد برامج تصميم الدوائر الالكترونية		
زمن التطبيـــــــق :		رقم التطبيـق :	14
العناصر الالكترونية المستخدمة :	- Header 1x14, 1x7 , 2x6,1X2 - C 0.1 UF X 3 - D 1N4001 X2 - LED X 2		- IC MIC5205-3.3 - IC ATMEGA328B - LED X2 - R 10K X 3 - SWITCH X 2 - ZTT 16 MHZ

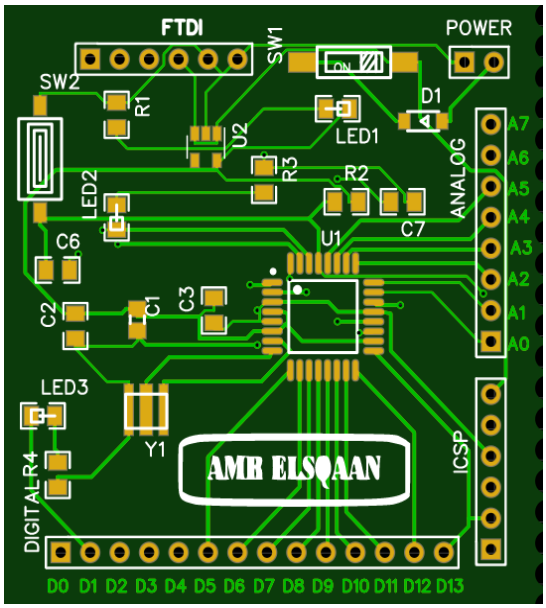
أ- المخطط النظري :



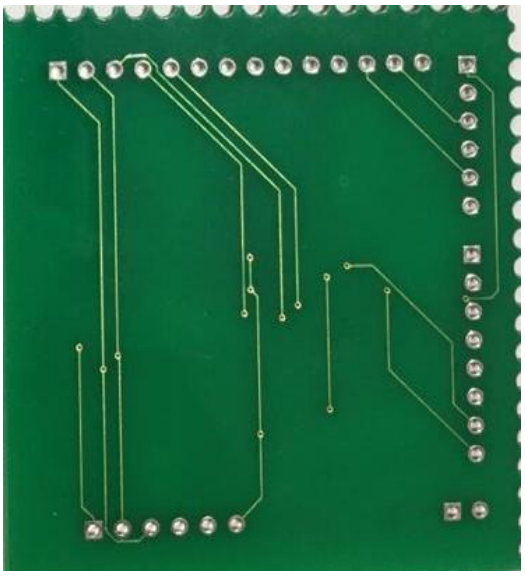
ب- لوحة PCB بعد تصميمها باستخدام أحد برامج تصميم الدوائر الإلكترونية :



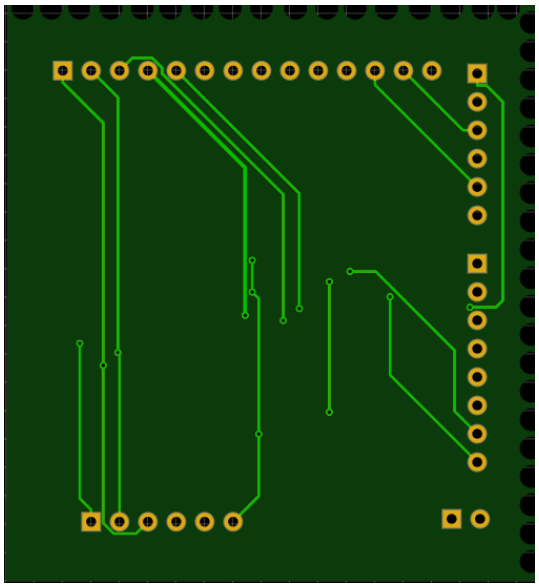
Electronic board



TOP Side



Electronic board



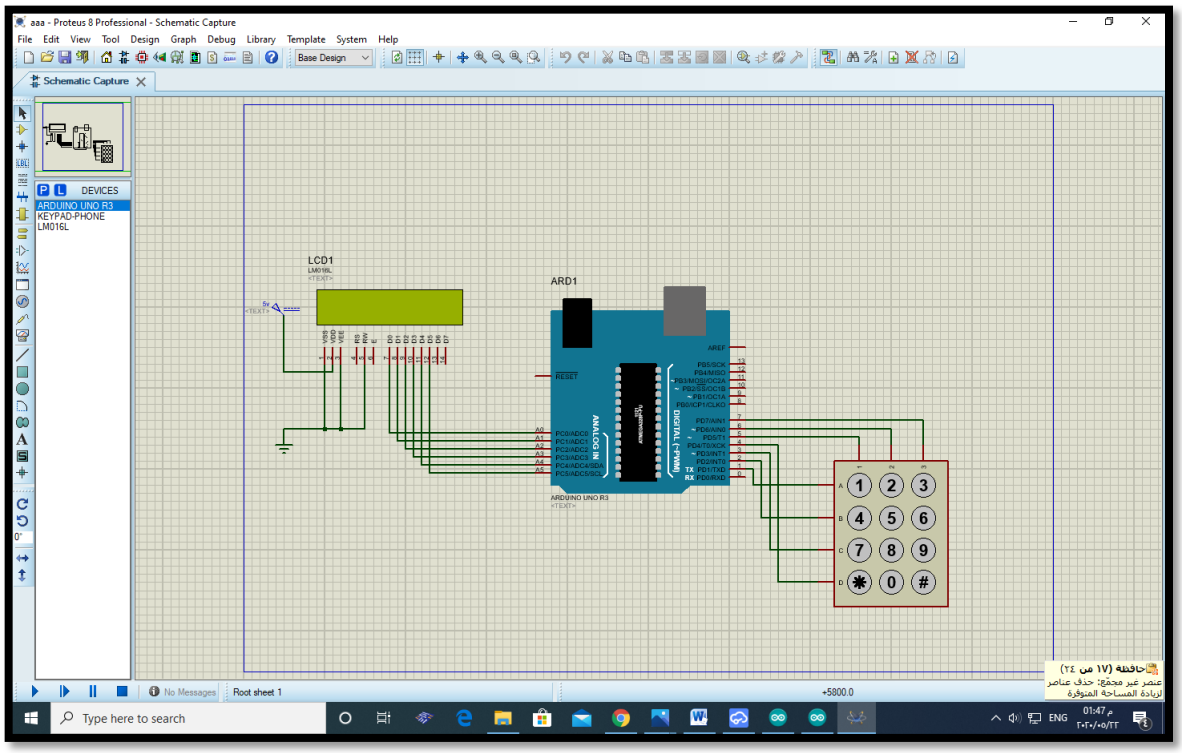
Bottom Side

عزيزي المتدرب:

تستطيع أن تنفذ التصميم الخاص بك ولا تتقيد بالتصميم الحالي فهو كدليل استرشادي فقط . .

المخرج التعليمي (3)	ينفذ منظومة تحكم مادية " Hard Wear " تعمل بالأردوينو		
معيـــــار الأداء (3)	ينفذ منظومة التحكم في قفل إلكتروني يعمل بالرقم السري .		
زمن التنفيـــــــــذ		أدلة التعليم	مرفق 8
معلومات إثرائيـــــه	موقع إلكتروني يقدم مجموعة من مشروعات الأردوينو ومن ضمنها مشروع قفل إلكتروني يعمل بالرقم السري		

أ- مخطط الدائرة

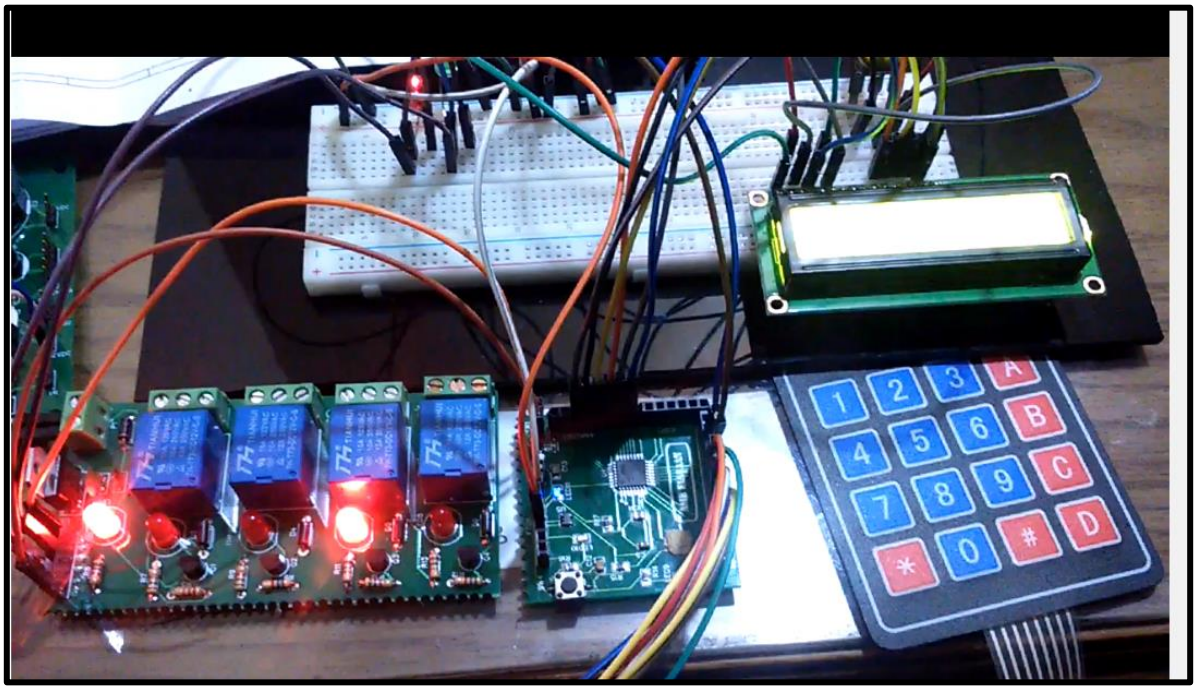



```
#include <LiquidCrystal.h> //include LCD library (standard library)
#include <Keypad.h> //include keypad library - first you must install library (library link in the video description)

#define redLED 10 //define the LED pins
#define greenLED 11

char* password ="1234"; //create a password
int pozisyon = 0; //keypad position

const byte rows = 4; //number of the keypad's rows and columns
const byte cols = 4;
```

```
char keyMap [rows] [cols] = {    //define the cymbols on the buttons of the keypad

 {'1', '2', '3', 'A'},
 {'4', '5', '6', 'B'},
 {'7', '8', '9', 'C'},
 {'*', '0', '#', 'D'}
} ;

byte rowPins [rows] = {1, 2, 3, 4}; //pins of the keypad
byte colPins [cols] = {5, 6, 7, 8};

Keypad myKeypad = Keypad ( makeKeymap(keyMap), rowPins, colPins, rows, cols);

LiquidCrystal lcd (A0, A1, A2, A3, A4, A5); // pins of the LCD. (RS, E, D4, D5, D6, D7)

void setup() {

 lcd.begin(16, 2) ;
 pinMode(redLED, OUTPUT);  //set the LED as an output
 pinMode(greenLED, OUTPUT);
 setLocked (true); //state of the password
}

void loop(){
 char whichKey = myKeypad.getKey(); //define which key is pressed with getKey
 lcd.setCursor(0, 0);
 lcd.print("    Welcome");
 lcd.setCursor(0, 1);
 lcd.print(" Enter Password");

 if(whichKey == '*' || whichKey == '#' || whichKey == 'A' ||      //define invalid keys
 whichKey == 'B' || whichKey == 'C' || whichKey == 'D'){

   pozisyon=0;
   setLocked (true);
   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("  Invalid Key!") ;
   delay(1000);
   lcd.clear();
 }
 if(whichKey == password [pozisyon]){
   pozisyon ++;
 }
 if(pozisyon == 4){
   setLocked (false);
   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("*** Verified ***") ;
   delay(3000);
   lcd.clear();
```

```
    lcd.setCursor(0, 0) ;
    lcd.print("  Mert Arduino");
    lcd.setCursor(0, 1);
    lcd.print("Tutorial Project");
    delay(7000);
    lcd.clear();
  }
  delay(100);
}

void setLocked(int locked){
  if(locked){
    digitalWrite(redLED, HIGH);
    digitalWrite(greenLED, LOW);
    }
    else{
      digitalWrite(redLED, LOW);
      digitalWrite(greenLED, HIGH);
    }
  }
```

ب- المخطط العملي للدائرة

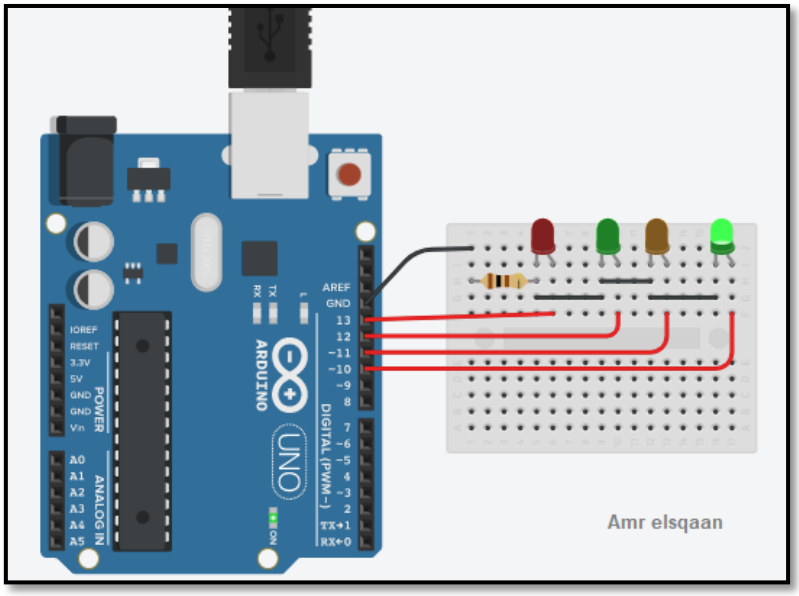


ملاحظة : هذه المنظومة استرشاديه ، يمكنك اختيار منظومة تحكم أخرى مع معلمك تخدم بيئتك ، كذلك يمكن اختيار العمل في مجموعة مكونة من ثلاث أفراد على الأكثر لإنجاز المنظومة (المشروع) مع عمل تقرير يوضح خطوات تنفيذ مشروعك

دليل حل التمارين

التمرين (1):

أ- مخطط التمرين



ب- الكود البرمجي

1

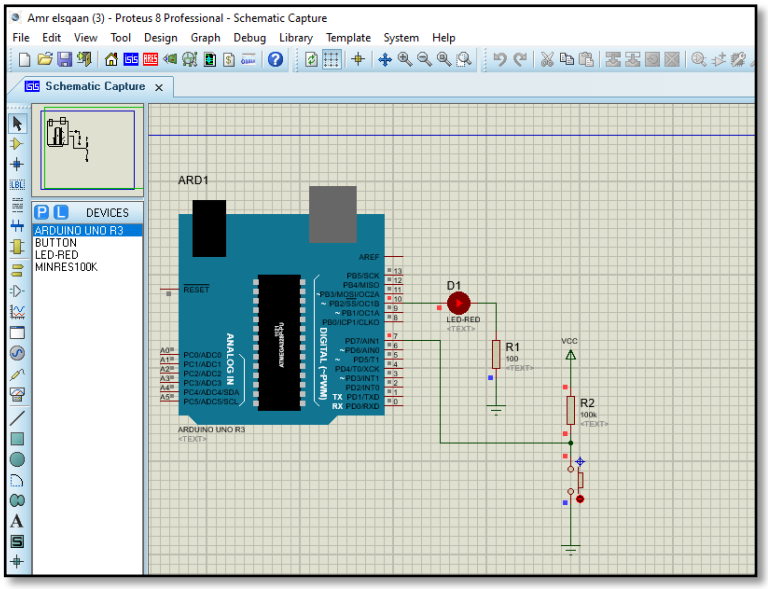
```
void setup()
{
  pinMode (13, OUTPUT);
  pinMode (12,OUTPUT);
  pinMode (11,OUTPUT);
  pinMode (10,OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
  digitalWrite(12, HIGH);
  delay(1000);
  digitalWrite(12, LOW);
  delay(1000);
  digitalWrite(11, HIGH);
  delay(1000);
  digitalWrite(11, LOW);
  delay(1000);
  digitalWrite(10, HIGH);
  delay(1000);
  digitalWrite(10, LOW);
  delay(1000);
}
```

2

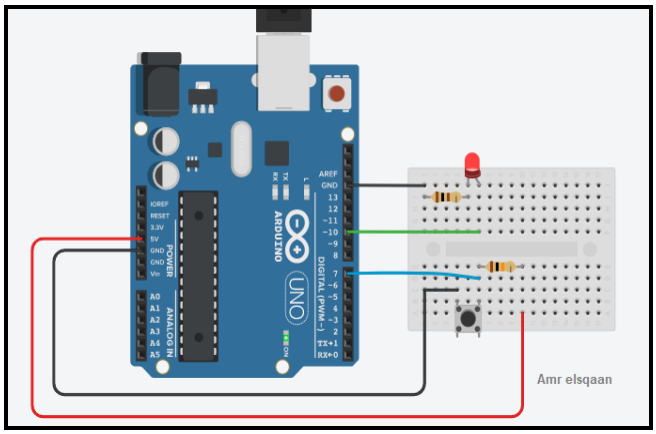
```
void setup()
{
  pinMode (13,OUTPUT);
  pinMode (12,OUTPUT);
  pinMode (11,OUTPUT);
  pinMode (10,OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(12, HIGH);
  delay(500);
  digitalWrite(12, LOW);
  delay(500);
  digitalWrite(11, HIGH);
  delay(500);
  digitalWrite(11, LOW);
  delay(500);
  digitalWrite(10, HIGH);
  delay(500);
  digitalWrite(10, LOW);
  delay(500);
}
```

التمرين (2):

أ- مخطط التمرين



ب-الكود البرمجي



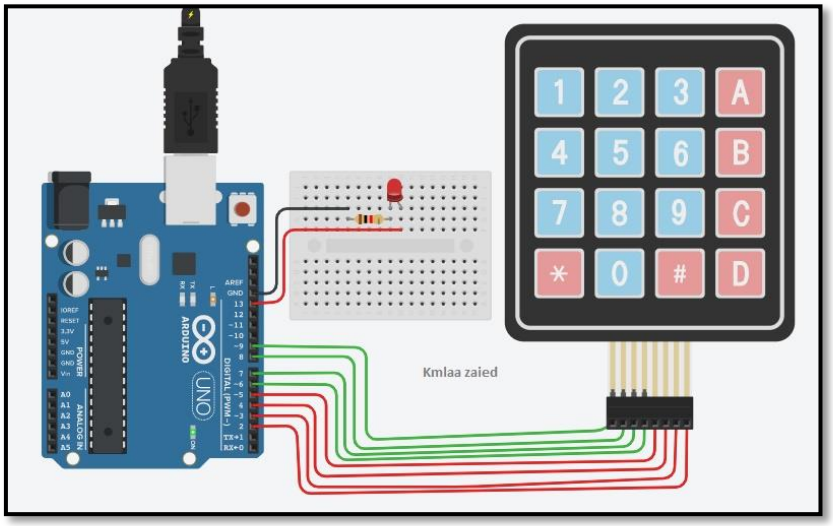

```
1  int ledPin = 10;
2  int buttonPin = 7;
3  int x = 0;
4
5  void setup()
6  {
7    pinMode(ledPin , OUTPUT);
8    pinMode(buttonPin ,INPUT);
9  }
10
11 void loop()
12 {
13   x = digitalRead(buttonPin);
14   if(x == HIGH)
15   {
16   digitalWrite(ledPin , HIGH);
17   delay(1000);
18   digitalWrite(ledPin , LOW);
19   delay(1000);
20   }
21   else
22    digitalWrite(ledPin , LOW);
23 }
```

شرح الكود :

	1
	2
	3
	4
	5
	6
	7

التمرين (3):

أ- مخطط التمرين



ب- الكود البرمجي :

```
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the key
byte colPins[COLS] = {5, 4, 3,2}; //connect to the column pinouts of the k

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
  pinMode(13,OUTPUT);
}

void loop(){
  char key = keypad.getKey();

  if (key=='*'){
    digitalWrite (13,HIGH);
  }
  if (key=='#'){
    digitalWrite (13,LOW);
  }
}
```

المراجع

1	كتاب احترف الأردوينو-القرية الهندسية (فريق معرض الازهر الشريف للتطبيقات الهندسية)
2	كتاب برمجة الأردوينو(م/ سامي قرامي)

3	الأردوينو كما لم تعرفه من قبل (م/ محمود مسلماني)
4	Arduino Learning
5	Arduino tutorial Rated Best
6	مواقع مختلفة من الانترنت