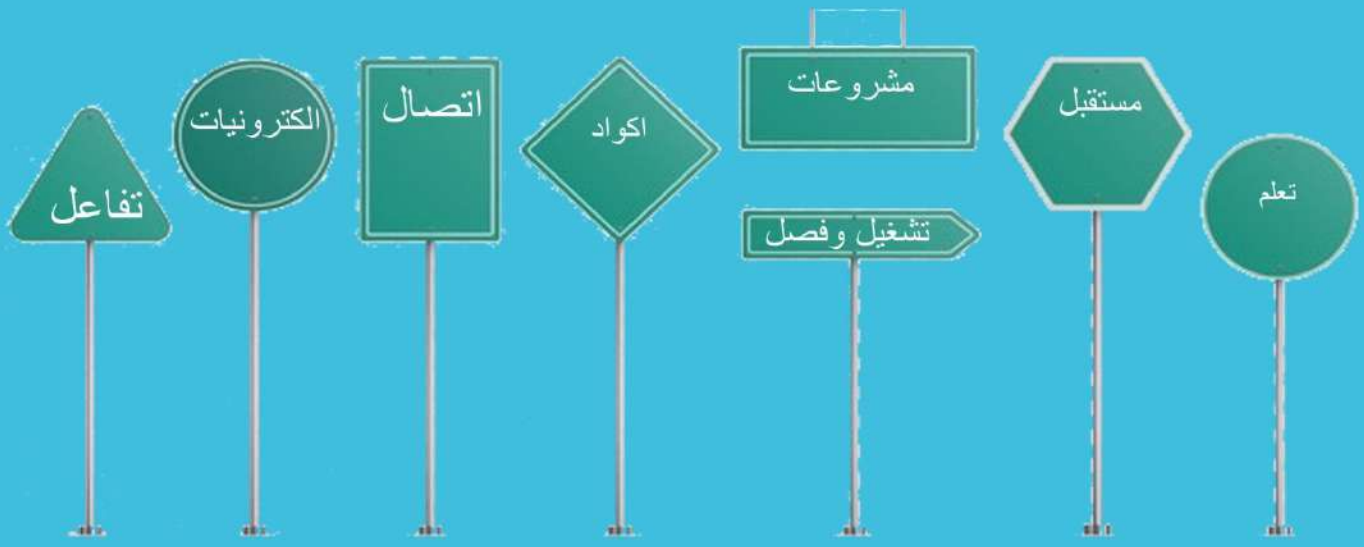
تعاليم الاردوينو



مهندس محمد محمود

تعاليم الاردوينو

م . محمد محمود

مقدمة الكتاب

يركز كتاب تعاليم الاردوينو على كيفية انشاء الدوائر الالكترونية بواسطة الاردوينو وكذلك كتابة الاكواد البرمجية اللازمة لتشغيل الاردوينو

كما ان الكتاب لا يركز على شرح العناصر الالكترونية بالتفصيل ولكنه يركز على كيفية انشاء مشروعك الخاص بأستخدام الاردوينو

حيث ان المستوى الكتاب للمبتدئين والراغبين فى تعلم الاردوينو

فقبل البدء فى تعلم الاردوينو عليك بتعلم وفهم العناصر والدوائر الالكترونية وكذلك كيفية كتابة وتعلم الاكواد البرمجية وخاصة لغة السى

واخيرا هذا الكتاب قمت بكتابتة بغرض التعليم وليس لأغراض تجارية فهو بين يدى القارىء بدون اى مقابل

واهلا لمن يرغب فى تقيم الكتاب واضافة الملاحظات على الايميل الاتى

mmmibrahima@gmail.com

محتويات الكتاب

ماهى جملة IF الشرطية

جملة switch case

البرنامج (٢) دائرة اردوينو تحتوى على مفتاحين وليد المفتاح الاول عند الضغط عليه يضىء الليد بينما عند الضغط على المفتاح الثانى يطفىء الليد

المراقب التسلسلىserial monitor

البرنامج (١) التحكم فى تشغيل واطفاء ليد عن طريق المراقب التسلسلى

البرنامج (٢) تتكون الدائرة من مفتاحين فقط فعند الضغط على اى مفتاح منهم يتم كتابة اسم المفتاح الذى تم الضغط عليه فى نافذة المراقب التسلسلى .

التعامل مع المدخلات التناظرية فى الاردوينو

البرنامج (١) نريد ان نجد قيمة التحويل بأستخدام مقاومة متغيرة قيمتها ٢.٢كيلو اوم وعرضها على المراقب التسلسلى ومقارنتها بجهاز قياس الجهد (الفولتميتر)

البرنامج (٢) عرض قيمة الجهد على اطراف المقاومة المتغيرة من خلال المراقب التسلسلى

الحساسات التناظرية analog sensor

البرنامج (١) قياس درجة الحرارة بأستخدام الحساس LM35 وعرض قيم درجات الحرارة فى المراقب التسلسلى

البرنامج (٢) قياس درجة الحرارة وعرضها فى المراقب التسلسلى مع ملاحظة انه عندما تكون درجة الحرارة اقل من او تساوى ٣٠ درجة يتم اضاءة الليد الاخضر اما اذا كانت درجة الحرارة اقل من او تساوى ٣٢ يتم اضاءة الليد الاحمر

المقاومة الضوئية photo resistor

البرنامج (٣) طباعة القيم التى تقرأها المقاومة الضوئية فى المراقب التسلسلى

البرنامج (٤) استشعار الضوء والظلام ففى حالة وجود ضوء يكتب الاردوينو فى المراقب التسلسلى light وفى حالة انعدام او انخفاض الضوء يتم طباعة dark

البرنامج (٥) استشعار الضوء والظلام ففى حالة وجود ضوء يكتب الاردوينو فى المراقب التسلسلى light مع اضاءة الليد الاحمر وفى حالة انعدام الضوء يتم طباعة dark مع اطفاء الليد الاحمر

حساسات قياس المسافات او الحساسات الفوق صوتية

البرنامج (٦) حساب المسافة بين الحساس والاجسام وطباعة المسافة فى المراقب التسلسلى بأستخدام حساس الموجات الفوق الصوتية

تعديل عرض النبضة

البرنامج (١) التحكم فى شدة اضاءة ليد بواسطة مقاومة متغيرة

البرنامج (٢) مفتاحين يتم التحكم من خلالهم فى شدة اضاءة الليد بحيث اذا تم الضغط على المفتاح الاول يزيد من شدة الاضاءة واذا تم الضغط على المفتاح الثانى يقلل من شدة الاضاءة

البرنامج (٣) التحكم فى شدة اضاءة ليد بواسطة مقاومة متغيرة بأستخدام دالة map

التحكم فى محركات التيار المستمر

طريقة تشغيل المحركات على لوحات الاردوينو

البرنامج (١) تشغيل واطفاء محرك بأستخدام مفتاحين احداهما للتشغيل والاخر للاطفاء مع ملاحظة وجود اثنين من الليد احداهما اخضر يضىء فى حالة تشغيل المحرك ويطفىء فى حالة اطفاء المحرك اما الليد الاحمر يضىء فى حالة اطفاء المحرك ويطفىء فى حالة تشغيل المحرك

كيفية عكس اتجاة دوران محرك تيار مستمر

البرنامج (٢) التحكم فى اتجاة دوران محرك حيث يتم التحكم بأستخدام ثلاث مفاتيح

الاول دوران المحرك فى اتجاة عقارب الساعة

الثانى دوران المحرك فى اتجاه عكس عقارب الساعة

الثالث ايقاف المحرك

السيرفو موتور

البرنامج (١) التحكم فى حركة سيرفو موتور مابين زوايا ٩٠ , ١٨٠ , ١٣٥, ٩٠ , ٠ , ٤٥ درجة

البرنامج (٢) التحكم فى زوايا دوران محرك سيرفو موتور عن طريق المراقب التسلسلى serial monitor

فعند كتابة ٠ يكون موضع زواية الدوران للمحرك هى ٠ درجة

اما عند كتابة ١ يكون موضع زواية الدوران للمحرك هى ٤٥ درجة

اما عند كتابة ٢ يكون موضع زواية الدوران للمحرك هى ٩٠ درجة

اما عند كتابة ٣ يكون موضع زواية الدوران للمحرك هى ١٣٥ درجة

اما عند كتابة ٤ يكون موضع زواية الدوران للمحرك هى ١٨٠ درجة

البرنامج (٣) التحكم فى دوران محرك سيرفو بواسطة مقاومة متغيرة

التعامل مع واجهات العرض

العارضة السباعية seven segment display

البرنامج (١) انشاء عداد تنازلى من ٩ الى ٠ باستخدام العارضة السباعية

البرنامج (٢) انشاء عداد تصاعدى من ٠ الى ٩ بالعارضة السباعية

التعامل مع الشاشات LCD

البرنامج (١) طباعة جملة hello world على السطر الاول اما على السطر الثانى طباعة thank you

البرنامج (٢) فى السطر الاول يتم طباعة الاعداد من ٠ الى ٩ ثم عمل ازاحة وطباعة للارقام مرة اخرى فى السطر الثانى من الشاشة

البرنامج (٣) طباعة الاعداد من ٠ الى ١٠٠ على الشاشة

البرنامج (٤) استخدام serial monitor فى ارسال بيانات الى شاشة LCDحيث السطر الاول يظهر فيه جملة pc connection

اما السطر الثانى يتم طباعة ما سيتم كتابتة فى شاشة المراقب التسلسلى

البرنامج (٥) يتم كتابة اى شيء على المراقب التسلسلى وارساله على الشاشة LCD بشرط عند وصول الحد الاقصى من الكلمات على الشاشة فيتم مسحها .

التعامل مع الاصوات

البرنامج (١) انشاء نغمة صوتية

التعامل مع لوحة المفاتيح keypad

البرنامج (١) عند الضغط على اى رقم من ارقام keypad يتم طباعة رقم المفتاح فى serial monitor

البرنامج (٢) انشاء برنامج يعمل على تشغيل محرك السيرفو بكلمة مرور فاذا كانت كلمة المرور صحيحة يتم اضاءة ليد اخضر وتشغيل محرك السيرفو اما اذا كانت كلمة المرور خاطئة يتم اضاءة الليد الاحمر ولا يتم تشغيل محرك السيرفو

التحكم باستخدام الاردوينو بواسطة البلوتوث

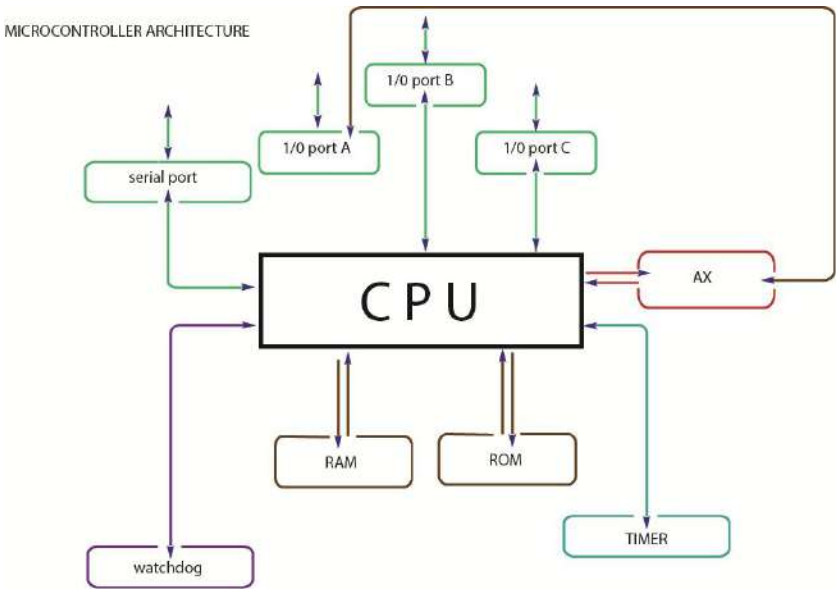
البرنامج (١) التحكم فى اضاءة واطفاء ليد بواسطة الكمبيوتر او الموبايل

مقدمة عامة

ماهو الميكروكنترولر ؟

الميكروكنترولر عبارة عن كمبيوتر صغير قابل للبرمجة ويحتوى على المكونات الاساسية الاتية :

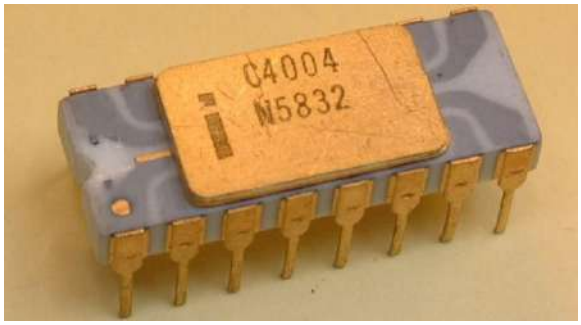
بروسيسور – ذاكرة – اجهزة دخل – اجهزة خرج



شكل يبين محتويات الميكروكنترولر

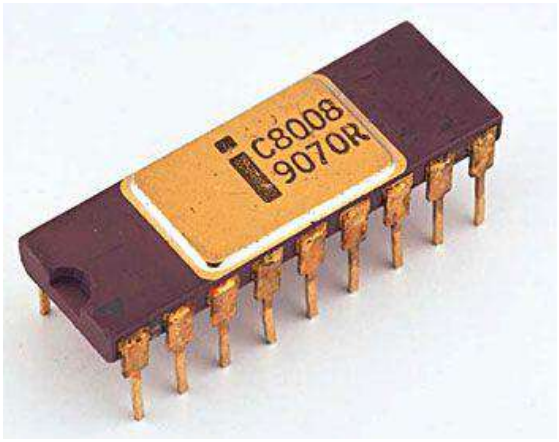
نبذة مختصرة عن تاريخ الميكروكنترولر :

ظهر اول ميكروبرسسور عام ١٩٧١ واطلق عليه ٤٠٠٤ ويدعم ٤ بت وكان من انتاج شركة انتل وتم استخدامة فى صناعة اول اله حاسبة محمولة



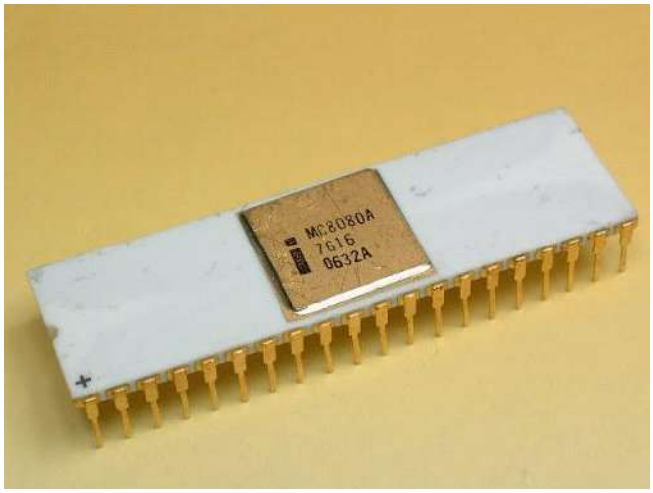
شكل ميكروبرسسور ٤٠٠٤

ثم ظهر بعد ذلك ميكروبرسسور ٨٠٠٨ ويدعم ٨ بت



شكل ميكروبرسسور ٨٠٠٨

ثم قدمت شركة انتل عام ١٩٧٤ الميكروبرسسور ٨٠٨٠ وتم استخدامه فى صناعة اول كمبيوتر شخصى

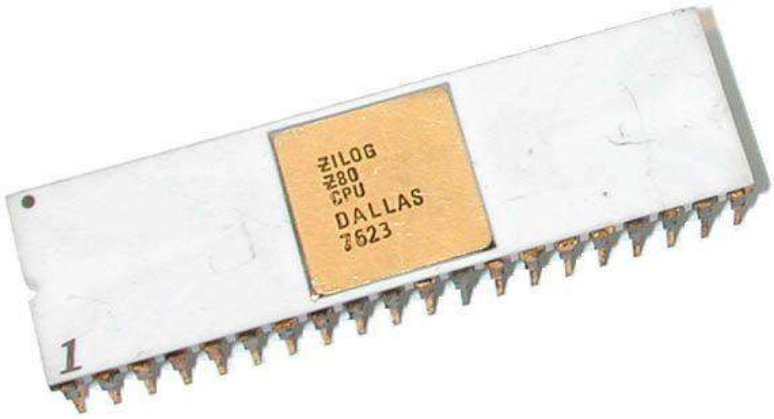


شكل يبين ميكروبرسسور ٨٠٨٠

ثم بدأت شركة موتورلا فى انتاج ميكروبرسسور ٦٨٠٠ ويدعم ٨ بت ويتميز بأنه يقدم مميزات عن ٨٠٨٠

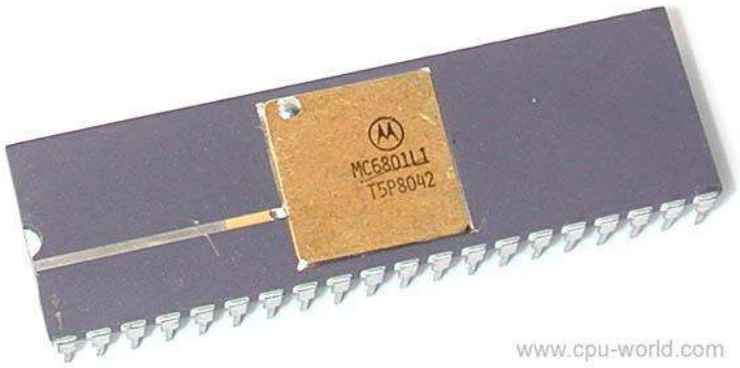
شكل يبين ميكروبرسسور ٦٨٠٠

بدأت شركة زى لوج بتقديم الميكروبرسسور Z80 وكان اكثر تقدما من الميكروبرسسور ٨٠٨٠



شكل يبين ميكروبرسسور Z80

ثم قدمت شركة موتورلا عام ١٩٧٦ الميكروبرسسور ٦٨٠١



شكل يبين ميكروبرسسور ٦٨٠١

ثم قدمت شركة انتل الميكروكنترولر ٨٠٤٨ ــ ٤٠٤٩ ــ ٨٠٥١

شكل يبين ميكروبروسسور ٨٠٥١

شكل يبين ميكروبرسسور ٨٠٤٨

ما المقصود بميكروبرسسور يدعم

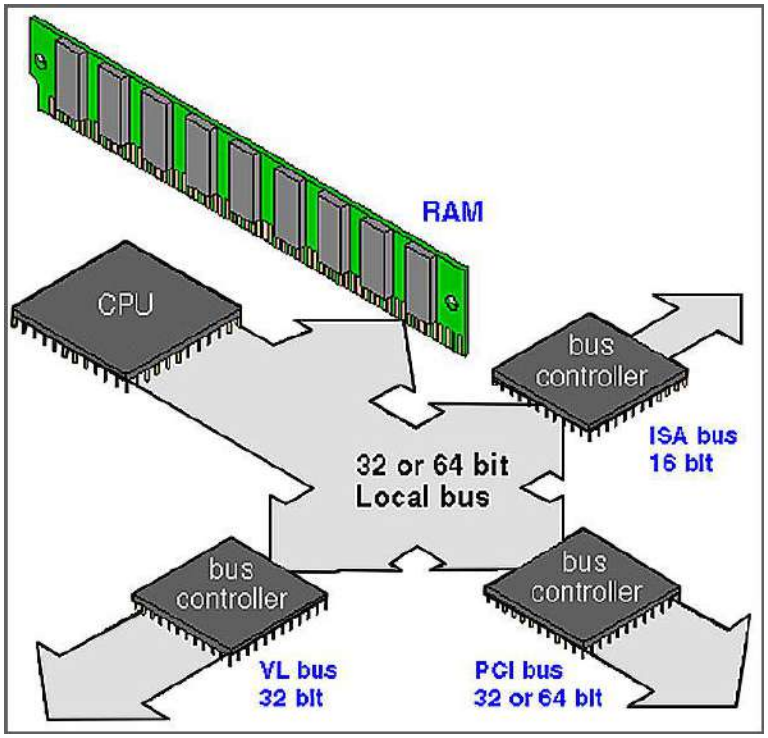
4 bit – 8 bit -16 bit – 32 bit – 64 bit

كلما زاد bit كلما زادت سعة البيانات التى يستطيع المعالج ان ينفذها وهذا يعنى زيادة فى سرعة وكفاءة الجهاز .

على سبيل المثال لنفترض ان هناك انبوب قطرة ١ بوصة وانبوب اخر قطره ٤ بوصة وخزان يحتوى على ٥٠ لتر من المياه فايهما سيقوم بسحب كمية مياة اكثر فى زمن اقل طبعا الانبوب الذى قطره ٤ بوصة كذلك المعالج الذى يدعم 32bit اسرع فى معالجة البيانات من المعالج الذى يدعم 16 bit .

فالمعالج 32 bit يستطيع ان يعالج بيانات موجودة بذاكرة سعة 4 GB بحد اقصى

بينما معالج 64 bit يستطيع ان يعالج بيانات موجودة بذاكرة قد تصل الى 32 GB



الشكل يبين ناقلات البيانات بسعة ٣٢ بت او بسعة ٦٤ بت

ماهو الاردوينو ؟

الاردوينو عبارة عن لوحة الكترونية مكونة من ميكروكنترولر ودائرة power وهى لوحة قابلة للبرمجة . يعتبر الاردوينو من اللوحات مفتوحة المصدر open source

يتوقف نوع لوحة الاردوينو على نوع الميكروكنترولر فعلى سبيل المثال :

اردوينو اونو تعتمد على ميكروكنترولر ATmega 328P

اردوينو ليوناردو تعتمد على ميكروكنترولر ATmega 32u4

ما المقصود ان الاردوينو مفتوح المصدر ؟

اى انه من الممكن الاطلاع على تصميم اللوحة والتعديل بها . كما يمكن الاطلاع على البرامج والتعديل بها . و يمكنك اختيار اى لوحة من لوحات الاردوينو من الجدول التالى :

Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB
Uno	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/6	1	2	32	Regular
Due	AT91SAM3X8E	3.3 V/7-12 V	84 Mhz	12/2	54/12	-	96	512	2 Micro
Leonardo	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro
Mega 2560	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular
Mega ADK	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular
Micro	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro
Mini	ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	1	2	32	-
Nano	ATmega168 ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	0.512 1	1 2	16 32	Mini-B
Ethernet	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/4	1	2	32	Regular
Esplora	ATmega32u4	5 V/7-12 V	16 Mhz	-	-	1	2.5	32	Micro
ArduinoBT	ATmega328	5 V/2.5-12 V	16 Mhz	6/0	14/6	1	2	32	-
Fio	ATmega328P	3.3 V/3.7-7 V	8 Mhz	8/0	14/6	1	2	32	Mini
Pro (168)	ATmega168	3.3 V/3.35-12 V	8 Mhz	6/0	14/6	0.512	1	16	-
Pro (328)	ATmega328	5 V/5-12 V	16 Mhz	6/0	14/6	1	2	32	-
Pro Mini	ATmega168	3.3 V/3.35-12 V 5 V/5-12 V	8 Mhz 16Mhz	6/0	14/6	0.512	1	16	-
LilyPad	ATmega168V ATmega328V	2.7-5.5 V/2.7-5.5 V	8 Mhz	6/0	14/6	0.512	1	16	-
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8 Mhz	4/0	9/4	1	2.5	32	Micro
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8 Mhz	4/0	9/4	1	2	32	-
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/2.7-5.5 V	8 Mhz	4/0	9/4	1	2	32	-

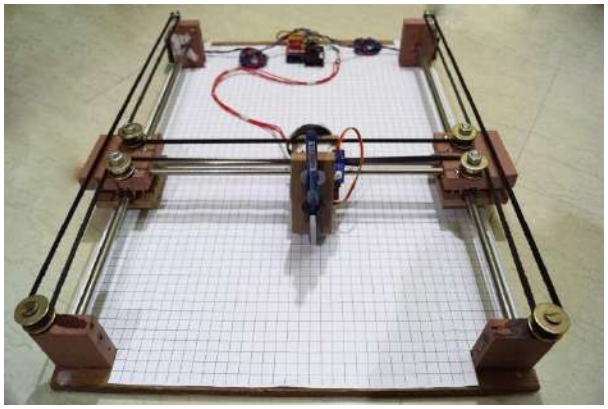
مميزات الاردوينو ؟

١. غير مكلف حيث يمكنك انشاء العديد من المشاريع بأقل التكاليف
٢. يمكن برمجتة فى اى نظام تشغيل سواء ويندوز او لينكس او ماكنتوش
٣. بسيط وغير معقد
٤. لدية دعم فنى قوى فى الحصول على الاستفسارات والاسئلة الخاصة بالمشاريع
٥. الاردوينو مفتوح المصدر سواء فى hardware / software

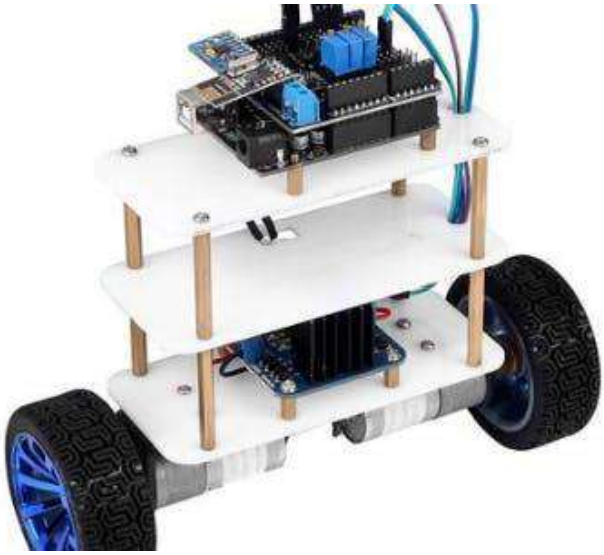
استخدامات الاردوينو :

من الممكن استخدم الاردوينو فى عمل اى مشروع مثل :

Drawing machine

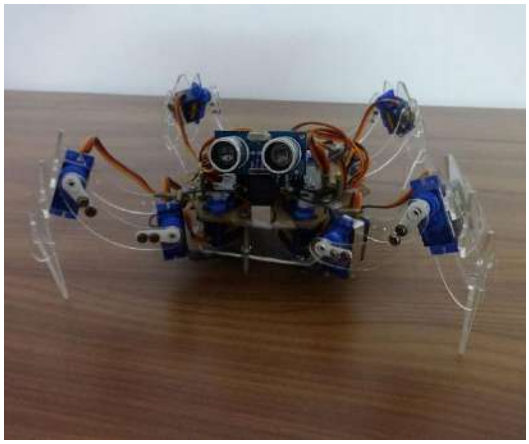


Balancing robot



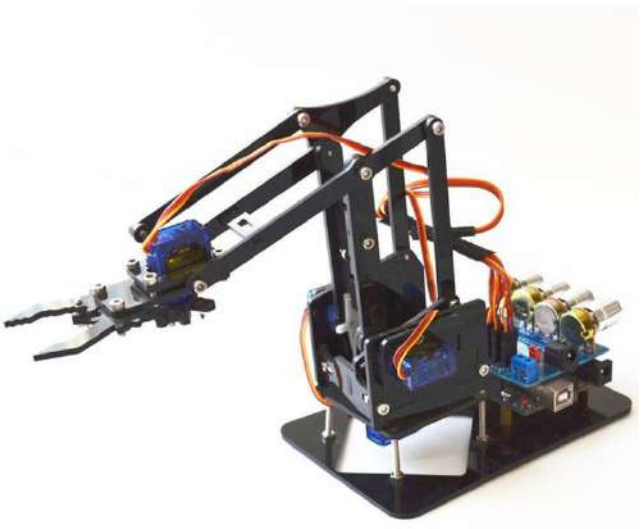
Smart watch

Spider quadruped robot



Quad copter

Arm robot



3d printer

انواع لوحات الاردوينو :

قبل البدء فى معرفة انواع لوحات الاردوينو علينا معرفة العوامل التى تدخل فى اختيار اللوحة عند انشاء المشاريع :

١. مقاس المشروع اذا كان صغير او كبير عندها ستختار لوحة كبيرة اوصغيرة
٢. مواصفات الميكروكنترولر من حيث مقاس الذاكرة وسرعة البروسيسور
٣. عدد نقاط الدخل او الخرج التى يحتاجها المشروع
٤. امكانية اضافة خصائص اخرى للمشروع مثل wifi – Bluetooth – Ethernet

يمكن تقسيم لوحات الاردوينو الى نوعين :

Boards – modules

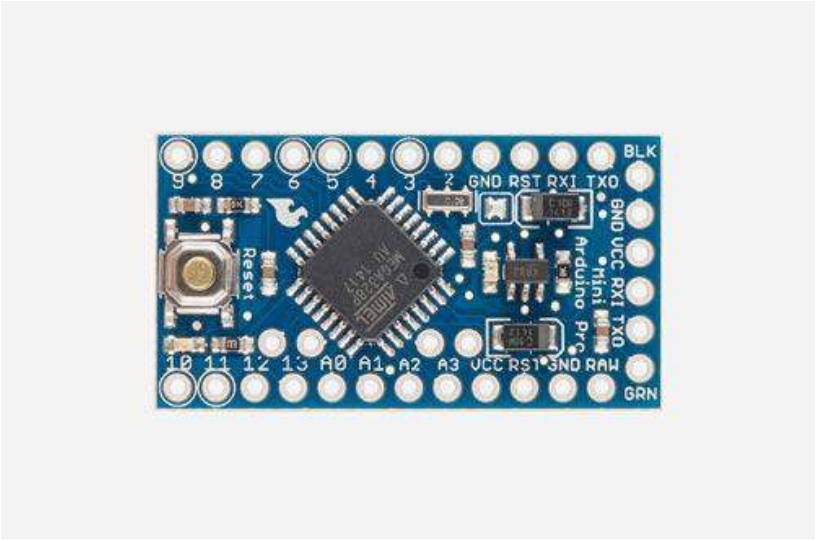
ما الفرق بين modules and boards

Boards : عبارة عن لوحة الكترونية يمكن اضافة الكروت عليها

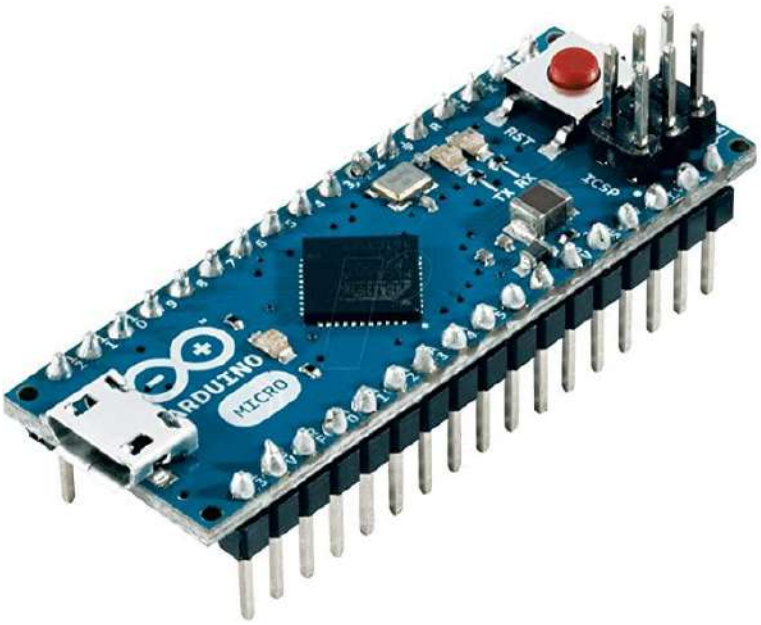
Modules : عبارة عن لوحة الكترونية صغيرة يتم تركيبها على لوحة اخرى لكى تؤدى وظيفة معينة .

انواع arduino modules :

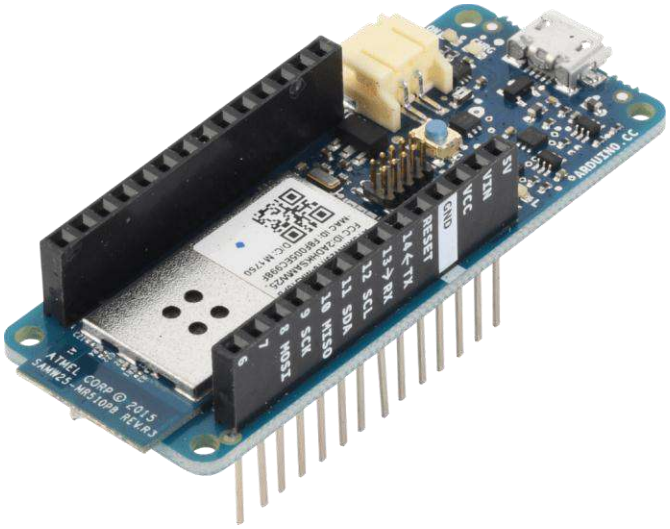
Arduino pro mini



Arduino micro



Arduino mkr1000

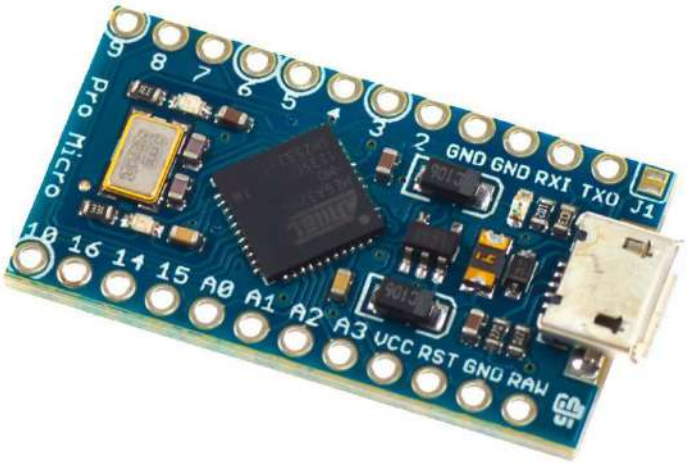


انواع arduino boards :

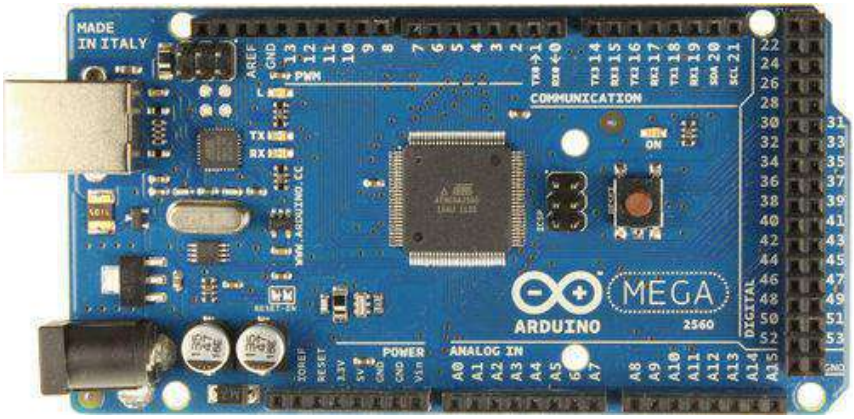
Arduino uno

Arduino 101

Arduino pro



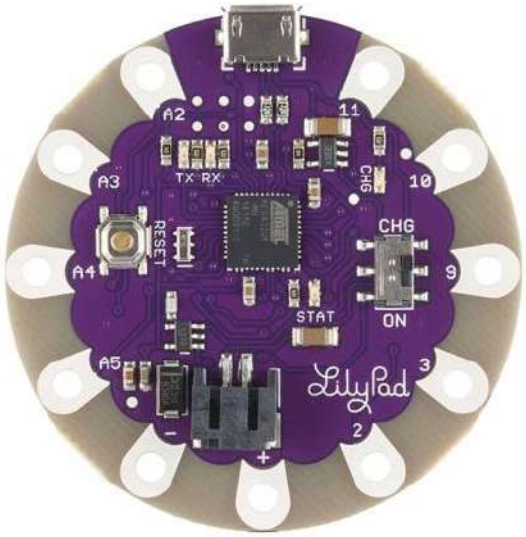
Arduino mega



Arduino zero

Arduino gemma

Lilypad arduino usb



اغطية او دروع الاردوينو arduino shields :

الاغطية او الدروع عبارة عن لوحة الكترونية فى نفس حجم الاردوينو ويتم تركيبها على لوحة الاردوينو للقيام بوظائف خاصة .

مميزاتها :

اولا : تجنب الاخطاء التى قد تحدث من تصميم الدائرة

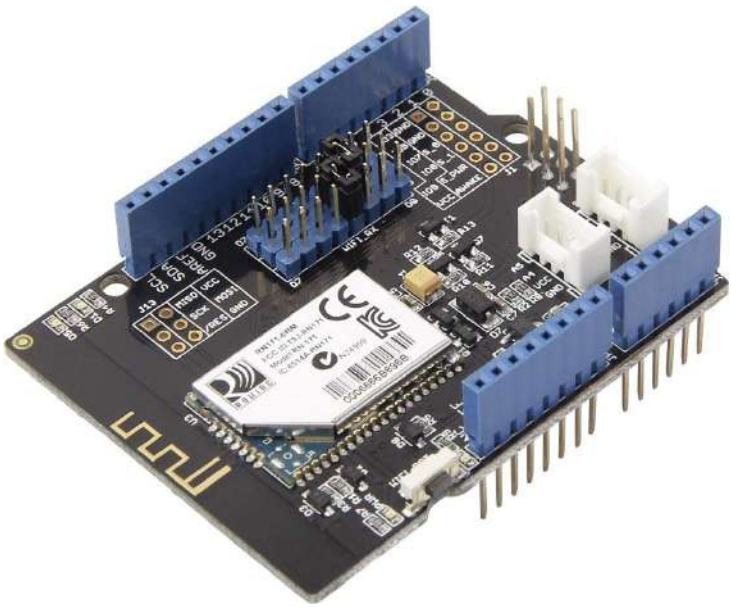
ثانيا : اختصار وتوفير الوقت والمجهود لبناء الدائرة

انواع دروع الاردوينو :

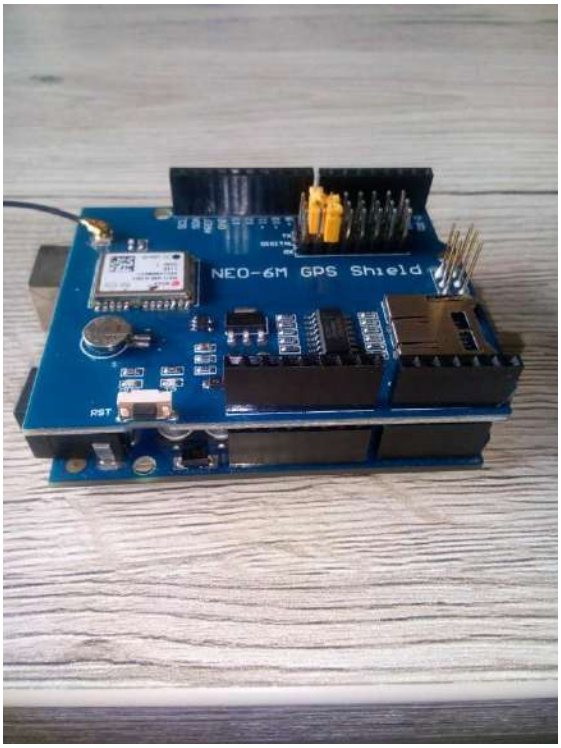
1. Ethernet , wifi , GPS shield
2. Display (LCD) , camera
3. Motor drives
4. Header , bread board shield

كما يمكنك صنع shield الخاص بك وتركيبها على لوحة الاردوينو

Ethernet shield arduino



wifi shield arduino



GPS shield arduino

شرح اطراف الاردوينو :

Arduino uno pin out

- Digital input / output pin

وهى الاطراف من ٠ الى ١٣ وتستخدم فى ادخال واخراج الاشارات الرقمية

- RX , TX pin

وتستخدم فى ارسال واستقبال البيانات بطريقة الاتصال التسلسلى serial communication

ولكن ما المقصود بال serial communication ؟

حيث يتم أرسال 1 bit كل فترة زمنية ويتم تحديدها بمعدل الانتقال buad rate

ونوع الاتصال هنا TTL serial Data

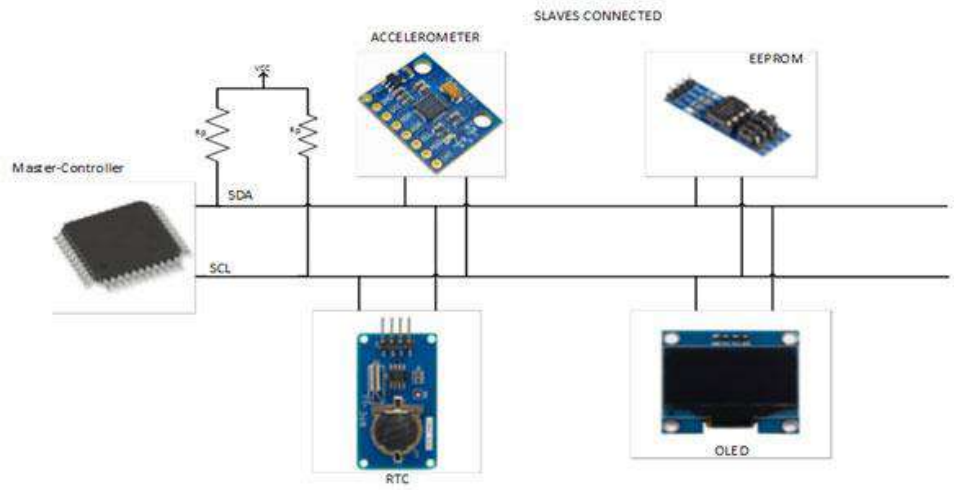
وتأخذ المصطلح التالى transistor – transistor Logic

- SDA , SCL pin

SDA ===> serial data

SCL ===> serial clock

وهو نوع من منافذ الاتصال ويستخدم طرفين فقط احداهما للأرسال والاخر للأستقبال ويطلق عليه اسم I2C communication اى two wire interface

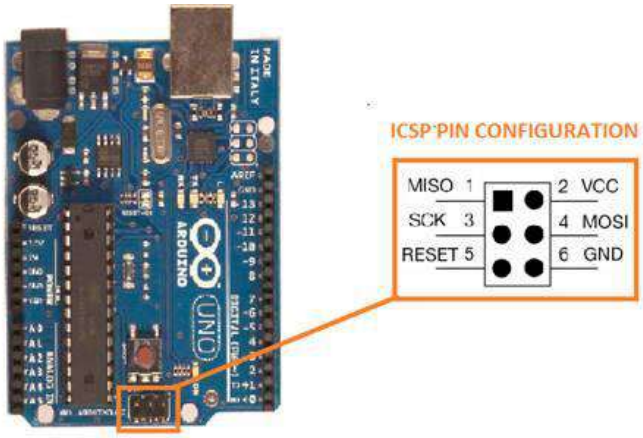


- PWM pin

اى تعديل نبضة الموجة Pulse Width Modulation ويمكن استخدامها للاطراف التى تأخذ علامة موج البحر وهى ٣ , ٥ , ٦ , ٩ , ١٠ , ١١ , ١٣

- ICSP pin

وهو اختصار ل in circuit serial programming وهو بروتكول الاتصال بين اكثر من ميكروكنترولر وبعضهم



ARDUINO UNO R3- ICSP PINS

- LED status

وتمثل حالة الميكروكنترولر على لوحة الاردوينو

وفيما يلى شرح لحالات led

يضىء عندما يكون المعالج فى استقبال بيانات RX ===>

يضىء عندما يكون المعالج فى حالة ارسال البيانات TX ===>

وهو متصل مع الطرف ١٣ ويعمل معه L ===>

حالة مصدر التغذية على اللوحة ON===>

- Analog input pin

وهى المسئولة عن قراءة الاشارة التناظرية والتى يتم قراءتها من الحساسات او المقاومات المتغيرة

ومن الاطراف التناظرية الموجودة على لوحة الاردوينو

A0 , A1 , A2 , A3 , A4 , A5

- AREF pin

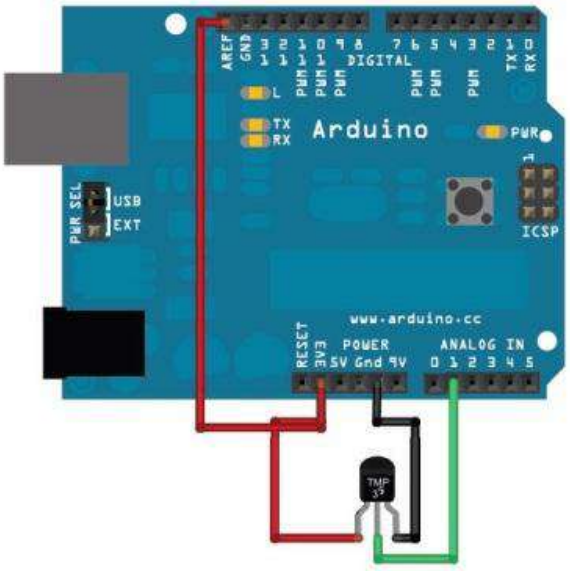
وهى اختصار لكلمة Analog Reference

يعمل Analog pin على جهد من ٠ الى ٥ فولت وتكون دقة التحويل فى analog digital converter هى 10 bit حيث يتم تحويل ٥ فولت الى ١٠٢٣ درجة ولكن

ماذا نفعل اذا كان الجهد هى 3.3 فولت بدلا من٥ فولت ؟

فى هذه الحالة سنستخدم AREF pin حيث يتم توصيل جهد 3.3 على AREF pin وبالتالى سيتغير الجهد من ٥ فولت الى ٣.٣ فولت .

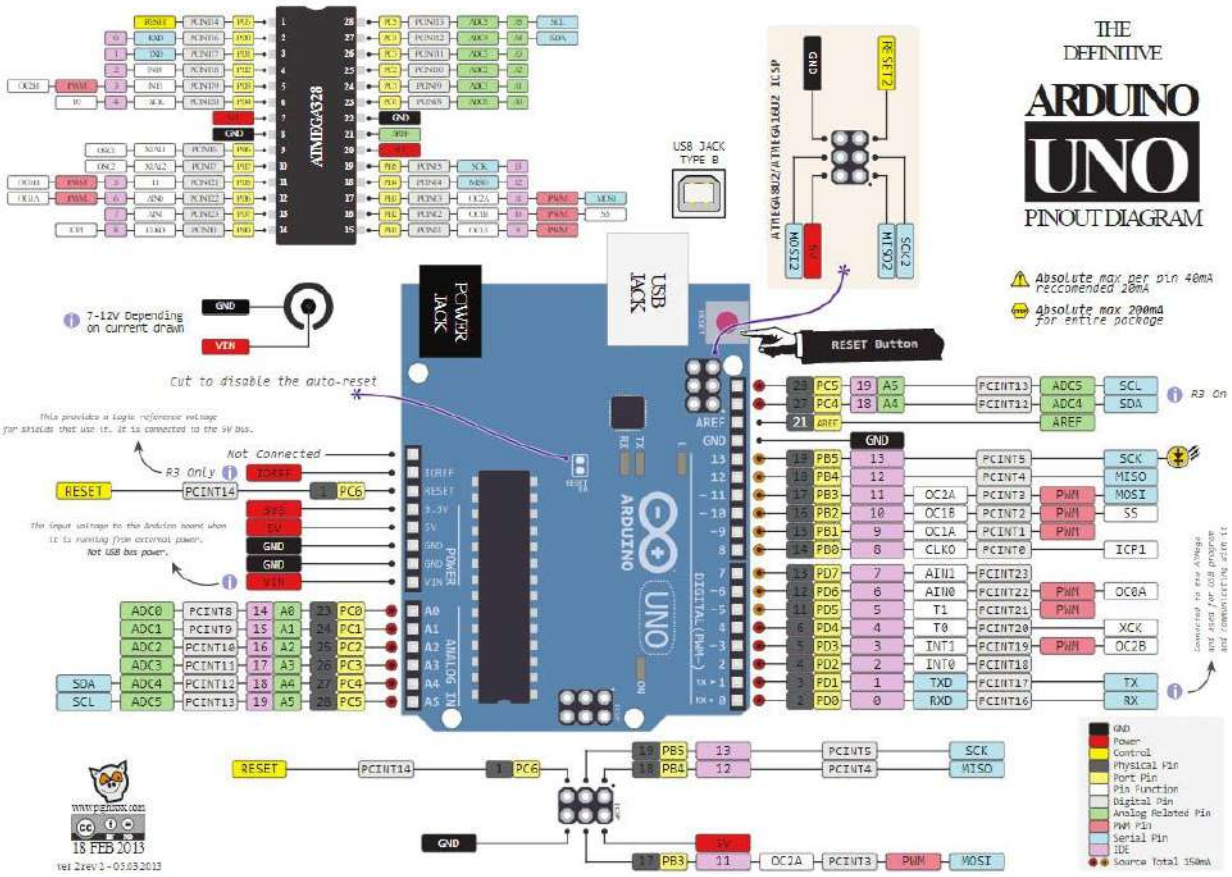
حيث سيتم تقسيم ٣.٣ فولت الى ١٠٢٣ درجة مما سيوفر لنا دقة عالية .ولا ننسى اننا سنكتب الكود الذى يعرف ان جهد المرجع هو ٣.٣ فولت .

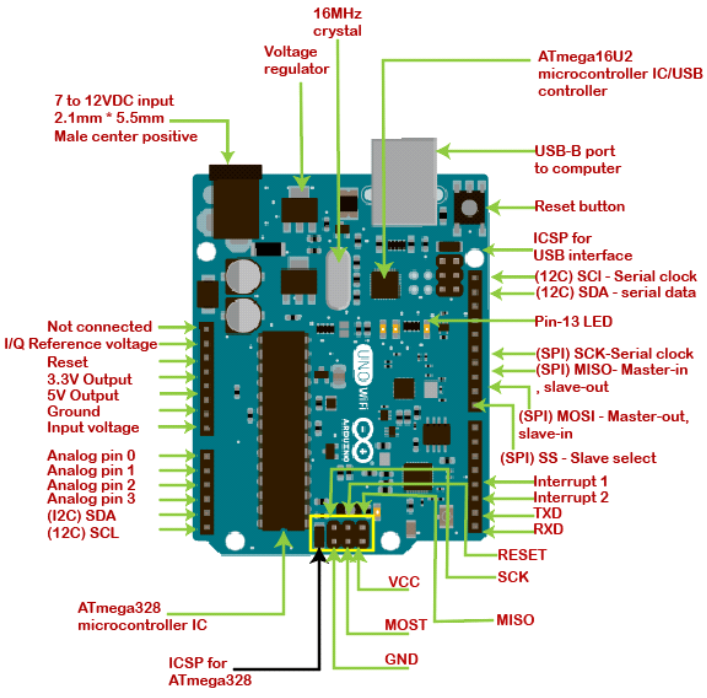


- Reset pin

ويستخدم لعمل اعادة تشغيل للميكروكنترولر ويعمل عندما تكون اشارتة LOW او GND

فيما يلى صورتين توضح arduino pinout بأختصار



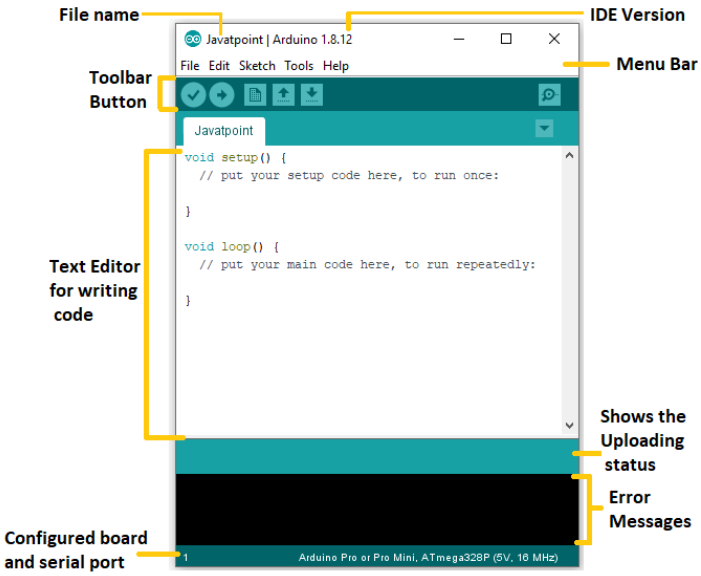


برنامج Arduino IDE

وهى اختصار لثلاث كلمات

Integrated development environment

اى بيئة التطوير المتكاملة تتكون واجهة الكتابة من الشكل التالى :



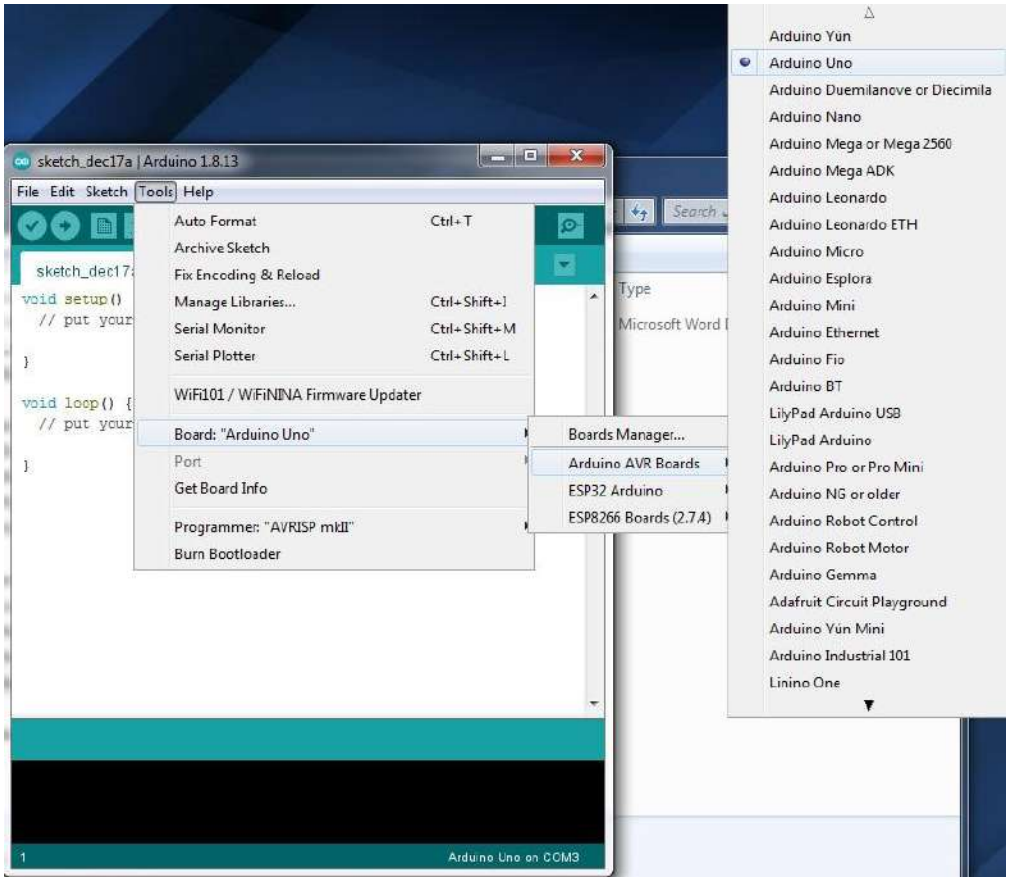
خطوات توصيل واعداد لوحة الاردوينو :

الخطوة الاولى : قم بتوصيل كابل USB بين الكمبيوتر ولوحة الاردوينو وبعد ثوانى يبدأ الويندوز بالتعرفة على لوحة الاردوينو .

الخطوة الثانية : قم بفتح Device manger ثم اختيار port (com & LPT) سوف تلاحظ ان الويندوز قد تعرف على الاردوينو واختار منفذ الاتصال بأسم Arduino uno (Com3)

الخطوة الثالثة : قم بفتح برنامج IDE لكى تختار نوع لوحة الاردوينو وذلك بأتباع المسار التالى :

Tools ---> board ---> arduino uno

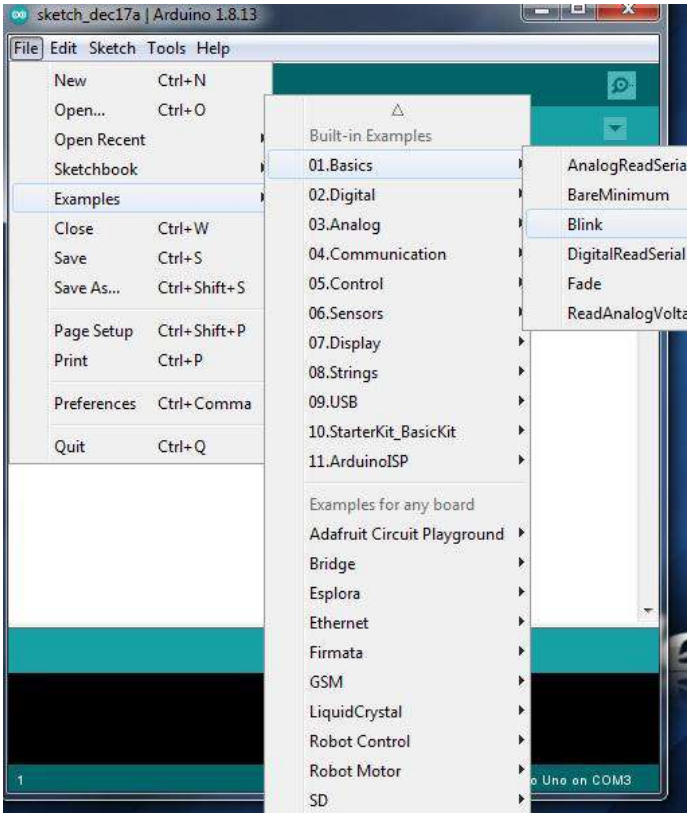


بعد تحديد نوع اللوحة يأتى هنا دور اختيار منفذ الاتصال وذلك بأتباع المسار التالى :

Tools ---> serial port ---> Com3

الخطوة الرابعة : نقوم بتحميل برنامج بسيط على لوحة الاردوينو من الامثلة الجاهزة وذلك للتأكد من صحة الاتصال بين الاردوينو والكمبيوتر وذلك بأتباع المسار التالى :

File ---> examples ---> basics ----> blink



التعرف على بيئة البرمجة فى الاردوينو :

اى ان قالب البرنامج يأخذ الشكل التالى :

Void setup ( )

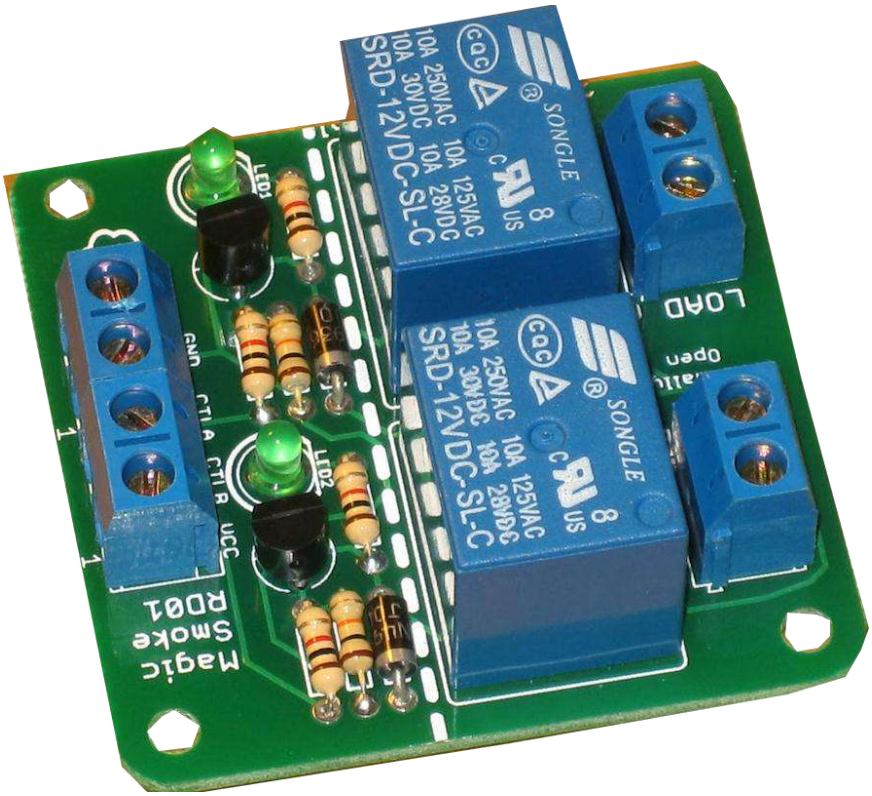
ويتم كتابة بها بعض التعريفات او المتغيرات وتعمل هذه الدالة مرة واحدة عند بدء التشغيل البرنامج

Void loop ( )

وهى دالة تعمل بأستمرار عند تنفيذ البرنامج

ويجب ان نراعى ان هناك بعض الاوامر يتم كتاباتها بالحروف الكبيرة والاخرى يتم كتابتها بالحروف الصغيرة والا سنرى ظهور الاخطاء قبل تحميل البرنامج المكتوب على الاردوينو.

التعامل مع اشارات الاخراج الرقمية فى الاردوينو



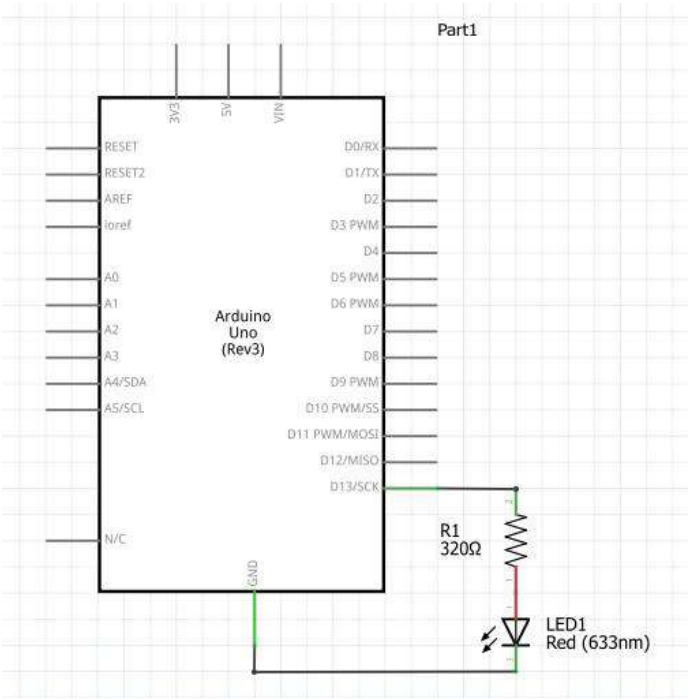
البرنامج (١)

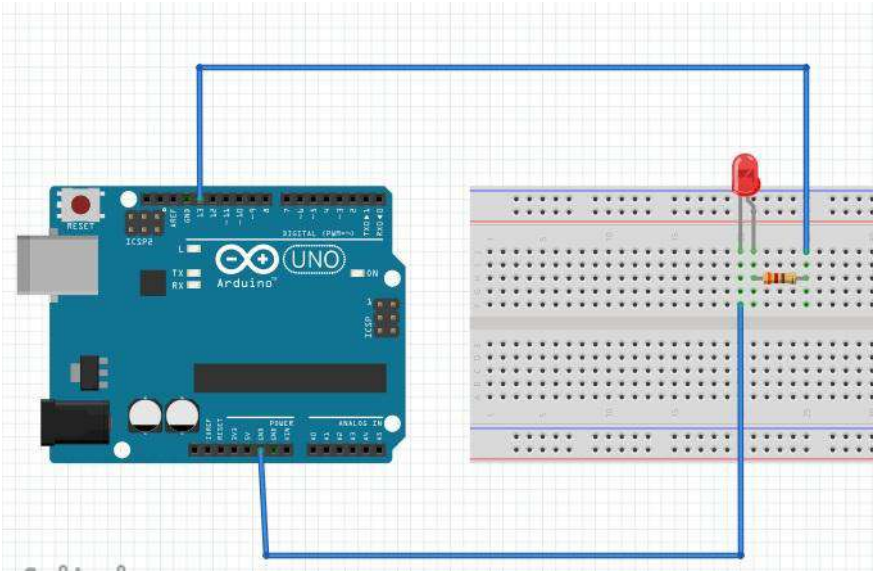
انشىء برنامج يقوم بتشغيل واطفاء الليد كل ثانية

المكونات الالكترونية :

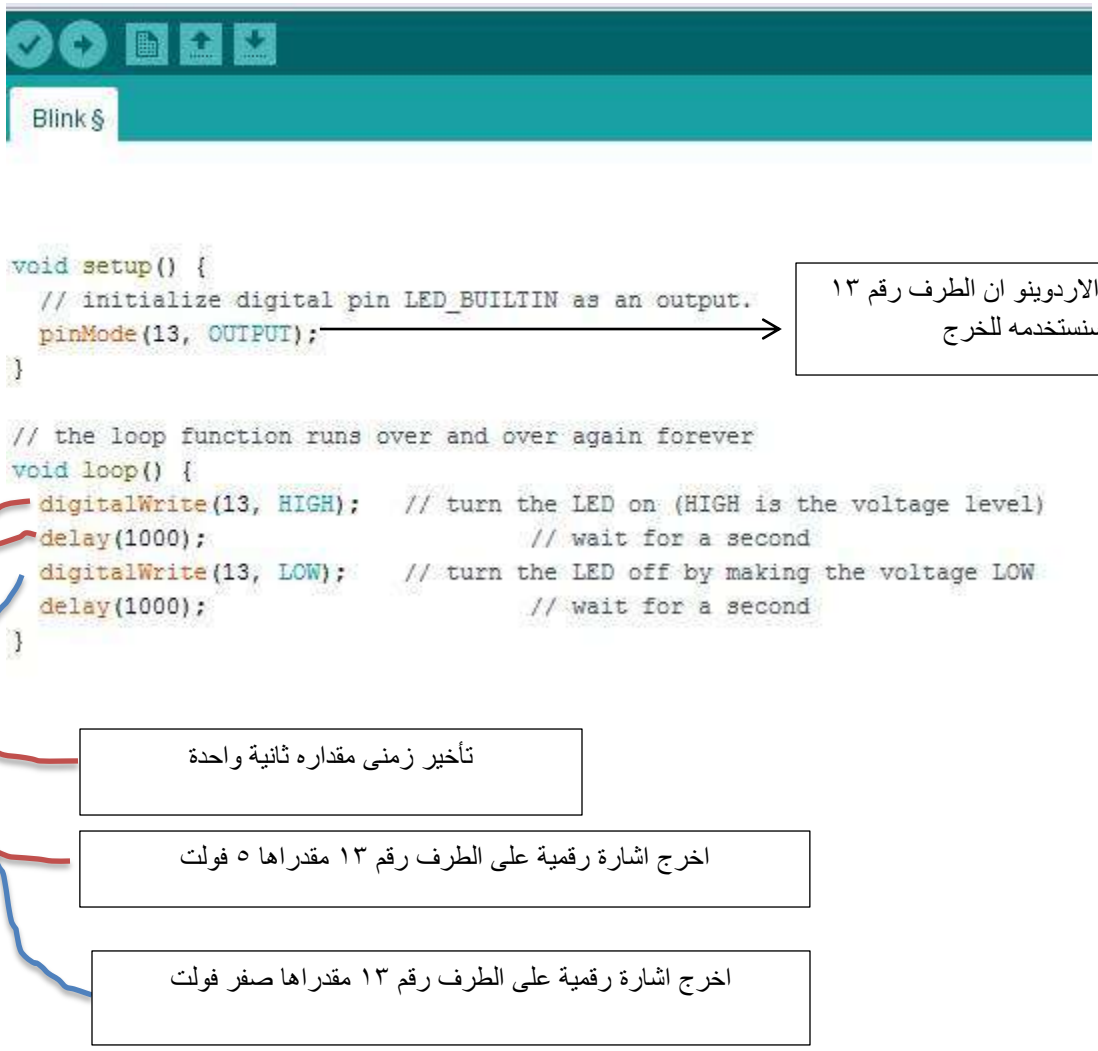
لوحة اختبار – ليد – اردوينو اونو – مجموعة اسلاك للتوصيل – مقاومة ثابتة ٢٢٠ اوم او ٣٢٠ اوم

دائرة التوصيل :





كود البرنامج :



ماذا تلاحظ ؟

سنلاحظ ان الليد يضىء ويطفىء كل ثانية

البرنامج (٢)

انشىء برنامج يقوم بتشغيل واطفاء الليد ٤ مرات مع العلم ان الزمن بين الاطفاء والتشغيل هو ١ ثانية وبعد ذلك يطفىء الليد لمدة ٣ ثوانى ثم يعود مرة اخرى لوضعة الابتدائى

فى هذه الحالة نستخدم امر التكرار for

ويأخذ الامر for الصورة التالية :

for (القيمة الابتدائية ; الشرط ; مقدار الزيادة)

{

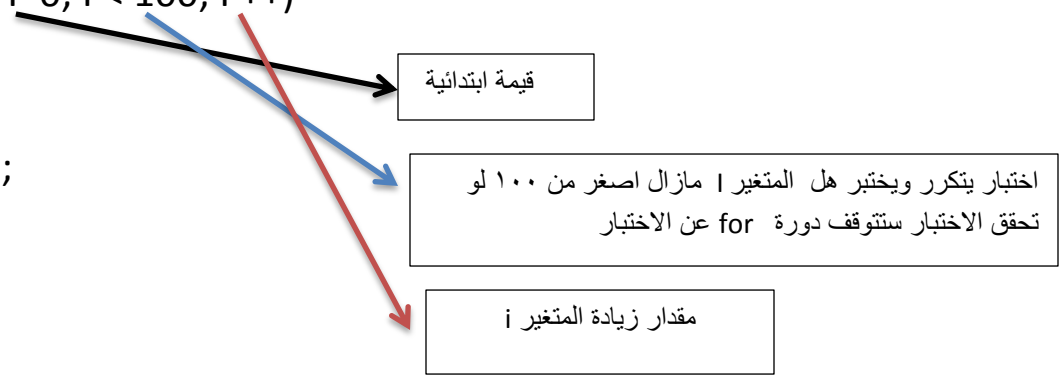
الحدث المراد تحقيقة

}

مع العلم انه عند تحقيق الحدث يتم الخروج عن الاقواس

مثال لكود مكتوب هو طباعة الارقام من ٠ الى ٩٩

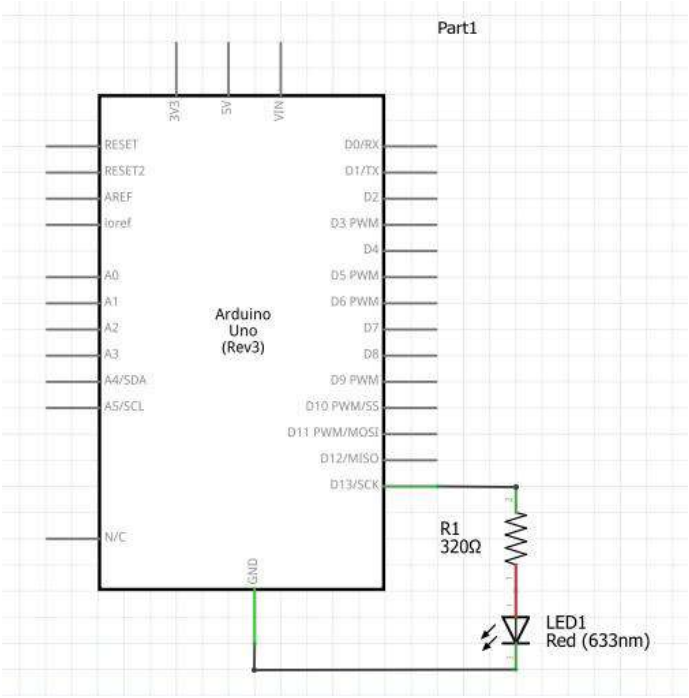
```
For (int I=0; I < 100; I ++)
{
Print (I);
}
```

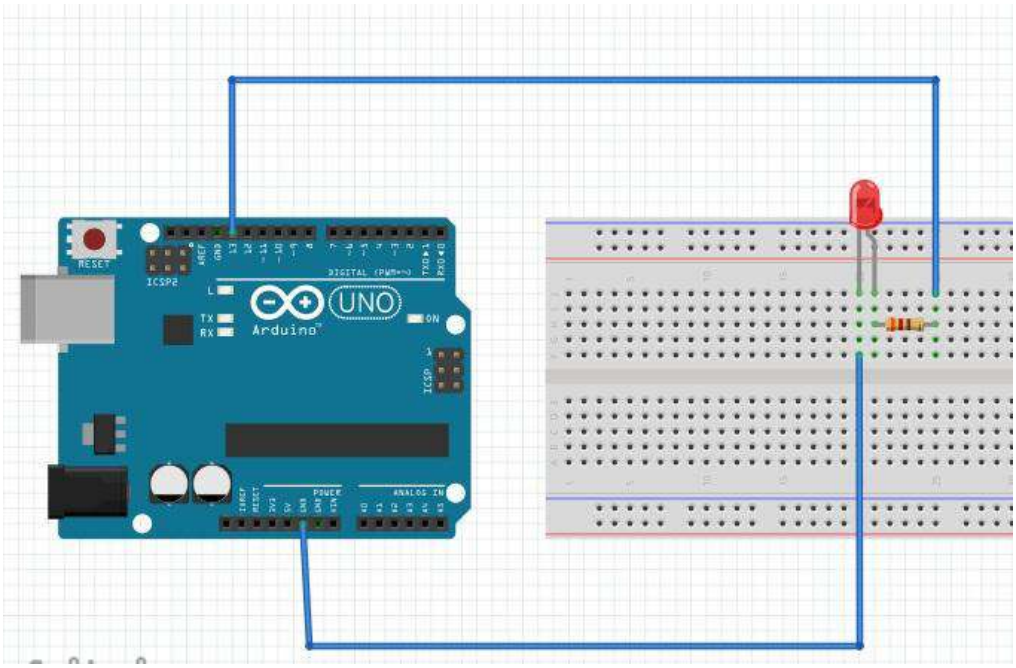


المكونات الالكترونية :

لوحة اختبار – ليد – اردوينو اونو – مجموعة اسلاك للتوصيل – مقاومة ثابتة ٢٢٠ اوم او ٣٢٠ اوم

دائرة التوصيل :





كود البرنامج :

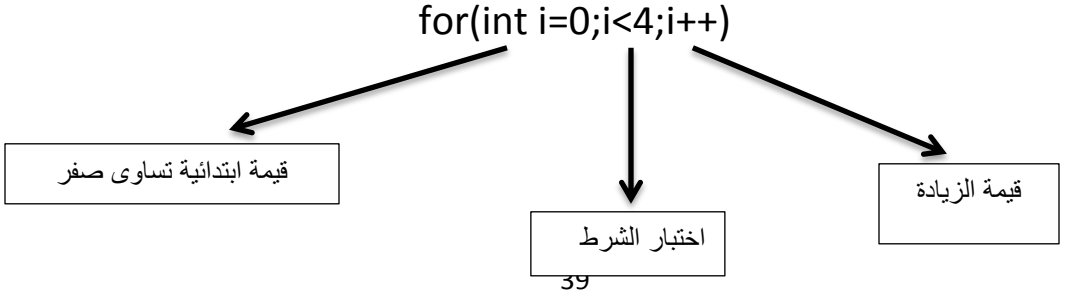
شرح البرنامج :

int pinled=13;

تعريف متغير صحيح اسمه pinled ويساوى القيمة ١٣

pinMode(pinled,OUTPUT);

for(int i=0;i<4;i++)



digitalWrite(pinled,HIGH);

اى اكتب على المتغير pinled اشارة ٥ فولت

digitalWrite(pinled,LOW);

اى اكتب على المتغير pinled اشارة ب ٠ فولت

delay(1000);

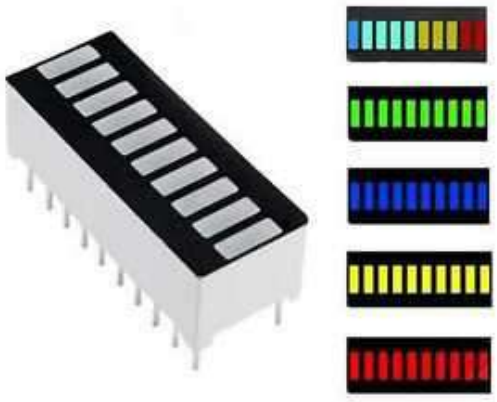
كود يتحكم فى التأخير الزمنى وهنا مقدارة ١ ثانية

القيمة ١٠٠٠ هنا تقدر ب ١٠٠٠ مللى ثانية اى تساوى ثانية واحدة

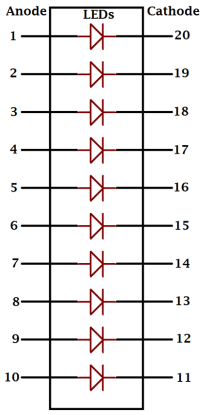
برنامج (٣)

انشىء برنامج يقوم بأنارة واطفاء LED BAR بالتسلسل

الشكل التالى يوضح LED BAR



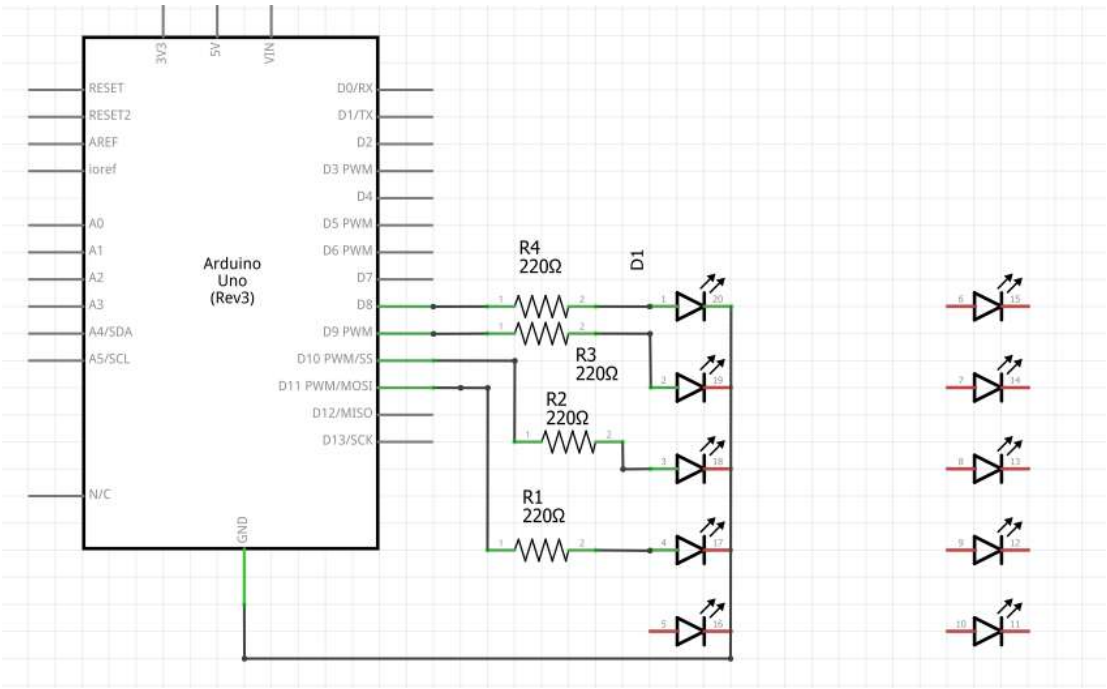
وهى مجموعة من LED توضع فى قالب بجانب بعضها كما بالشكل التالى :

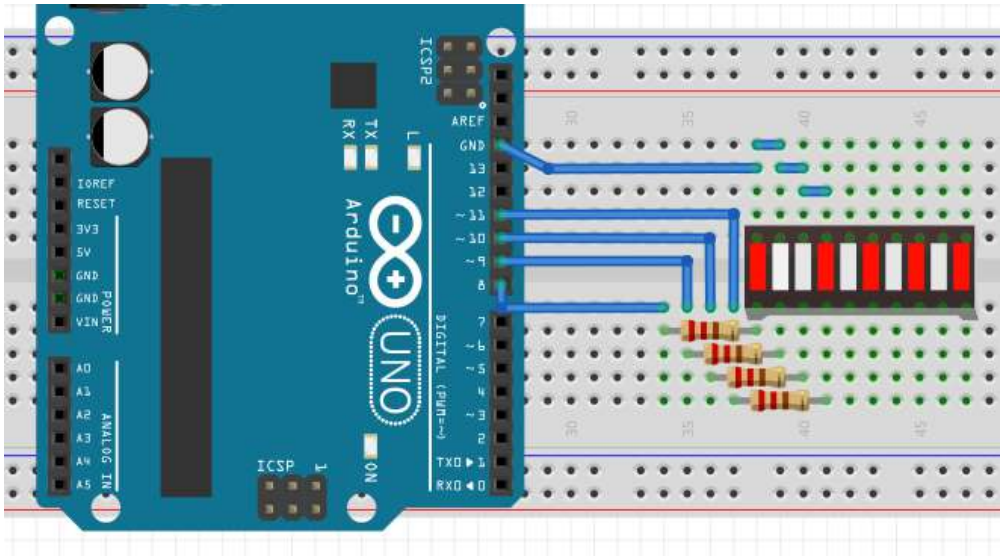


المكونات الالكترونية :

لوحة اختبار – LED BAR – اردوينو اونو – مجموعة اسلاك للتوصيل – مجموعة مقاومات ثابتة ٢٢٠ اوم او ٣٢٠ اوم

دائرة التوصيل :





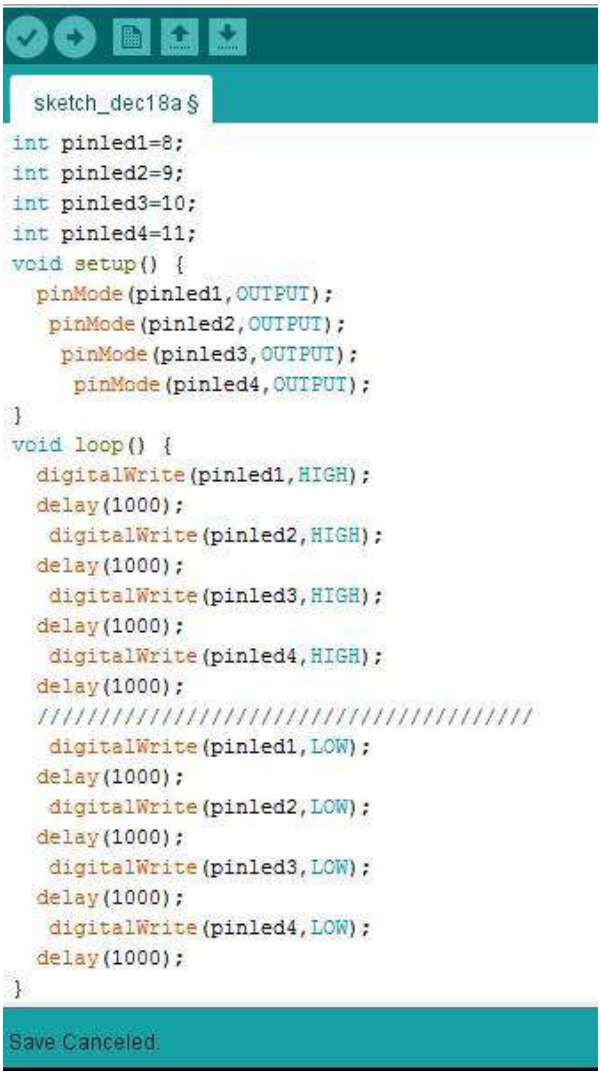
كود البرنامج :

سيتم كتابة البرنامج بطريقتين

الطريقة الاولى وهى تأخذ عدد من الاسطر فى الاكواد وهى الطريقة المباشرة

الطريقة الثانية وفيها يتم استخدام امر التكرار for والمصفوفة array والتى سيتم شرحها فيما بعد

اولا كود الطريقة الاولى :



ثانيا : كود الطريقة الثانية :

وفى هذه الطريقة سنستخدم المصفوفة array

فالمصفوفة هى عبارة عن قائمة تسمح بتخزين متغيرات من نفس النوع وتستعمل المصفوفة لأدارة مجموعة كبيرة من البيانات لها نفس النوع بأستخدام اسم واحد .

والمثال التالى يوضح طريقة الاعلان عن المصفوفة

لنفرض ان هناك مصفوفة اسمها number وتحتوى على اربع ارقام صحيحة هما

الرقم الاول يساوى ٣

الرقم الثانى يساوى ٦

الرقم الثالث يساوى ٩

الرقم الرابع يساوى ١٢

فيتم كتابة المصفوفة بالصيغة التالية :

Int number [4]={3,6,9,12}

فيتم تمثيل كل عدد على حده بالصيغة التالية :

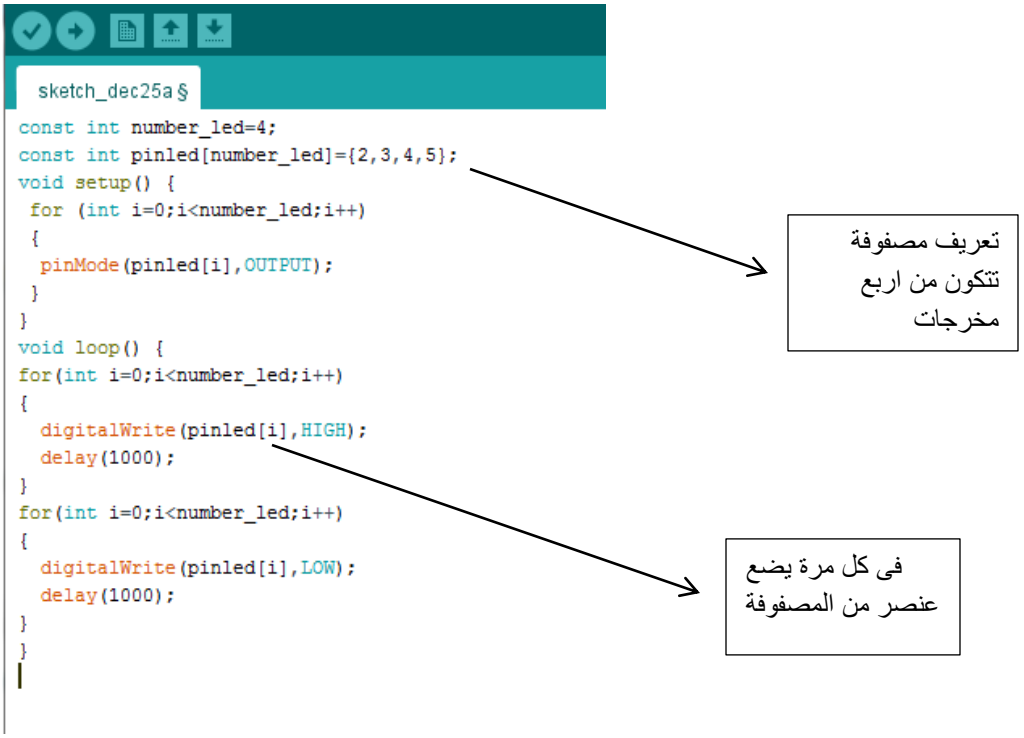
Number [0]=3 اى ان العنصر الاول من المصفوفة يساوى العدد ٣

Number [1]=6 اى ان العنصر الثانى من المصفوفة يساوى العدد ٦

Number [2]=9 اى ان العنصر الثالث من المصفوفة يساوى العدد ٩

Number [3]=12 اى ان العنصر الرابع من المصفوفة يساوى العدد ١٢

وبالتالى سيتم كتابة كود المثال السابق بأستخدام المصفوفة



التعامل مع اشارات الادخال الرقمية فى الاردوينو

التعامل مع المدخلات الرقمية digital input

والمقصود هنا هو كيف يمكن قراءة اشارة رقمية من دخل الاردوينو

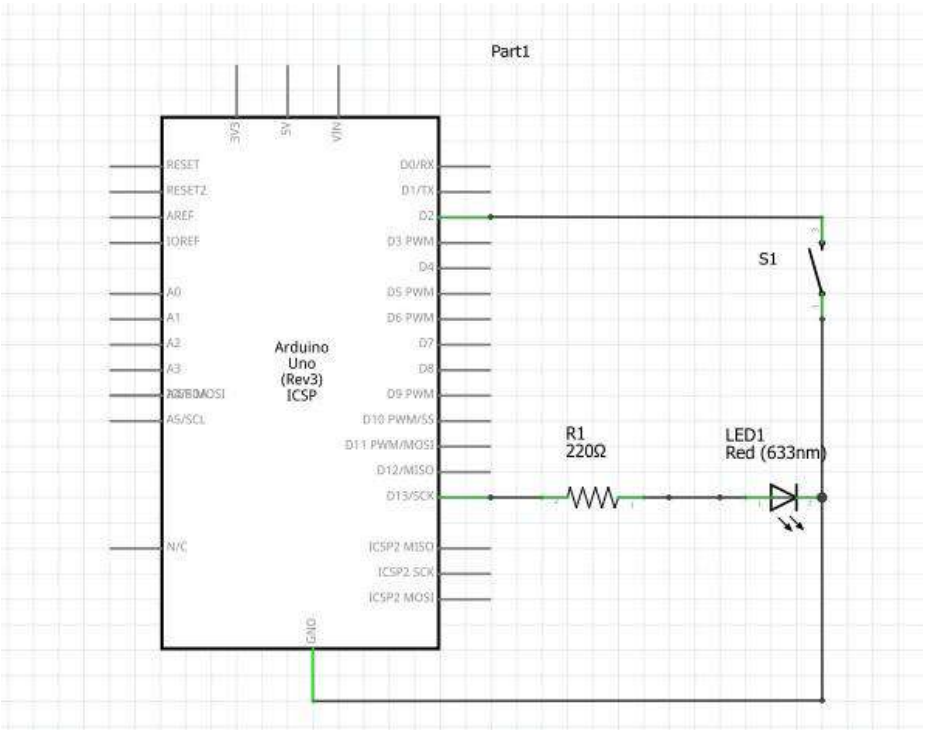
ويمكن تقسيم الاشارة الرقمية الى اشارة high وتمثل ٥ فولت اما الاشارة low فتمثل ٠ فولت

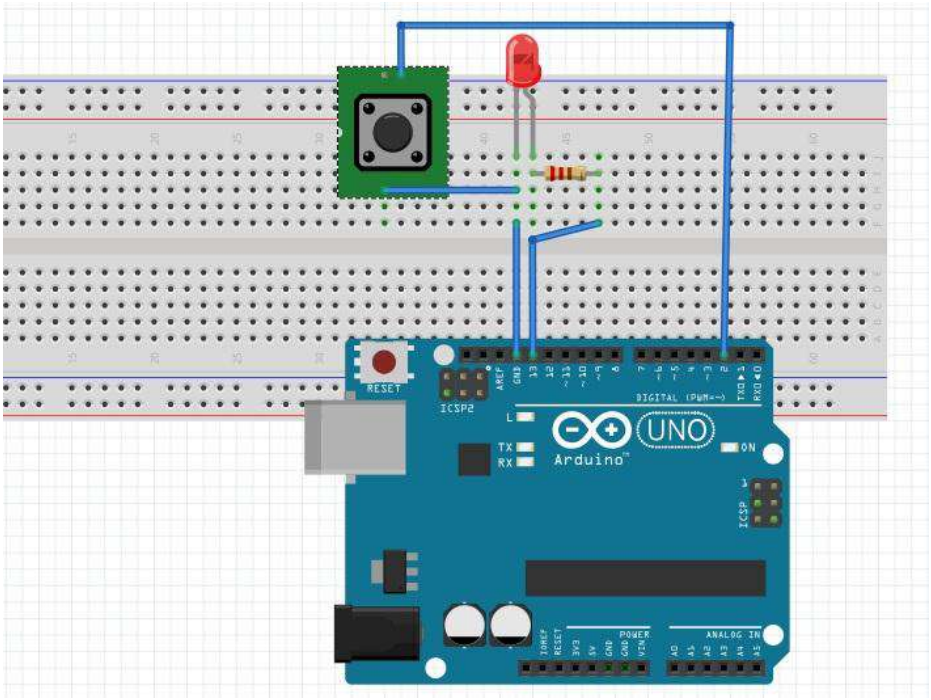
ويمكن تمثيل الاشارات الرقمية بمفتاح pushbutton

البرنامج (١)

انشىء برنامج يقوم بأنارة ليد عندما يتم الضغط على المفتاح

دائرة التوصيل :

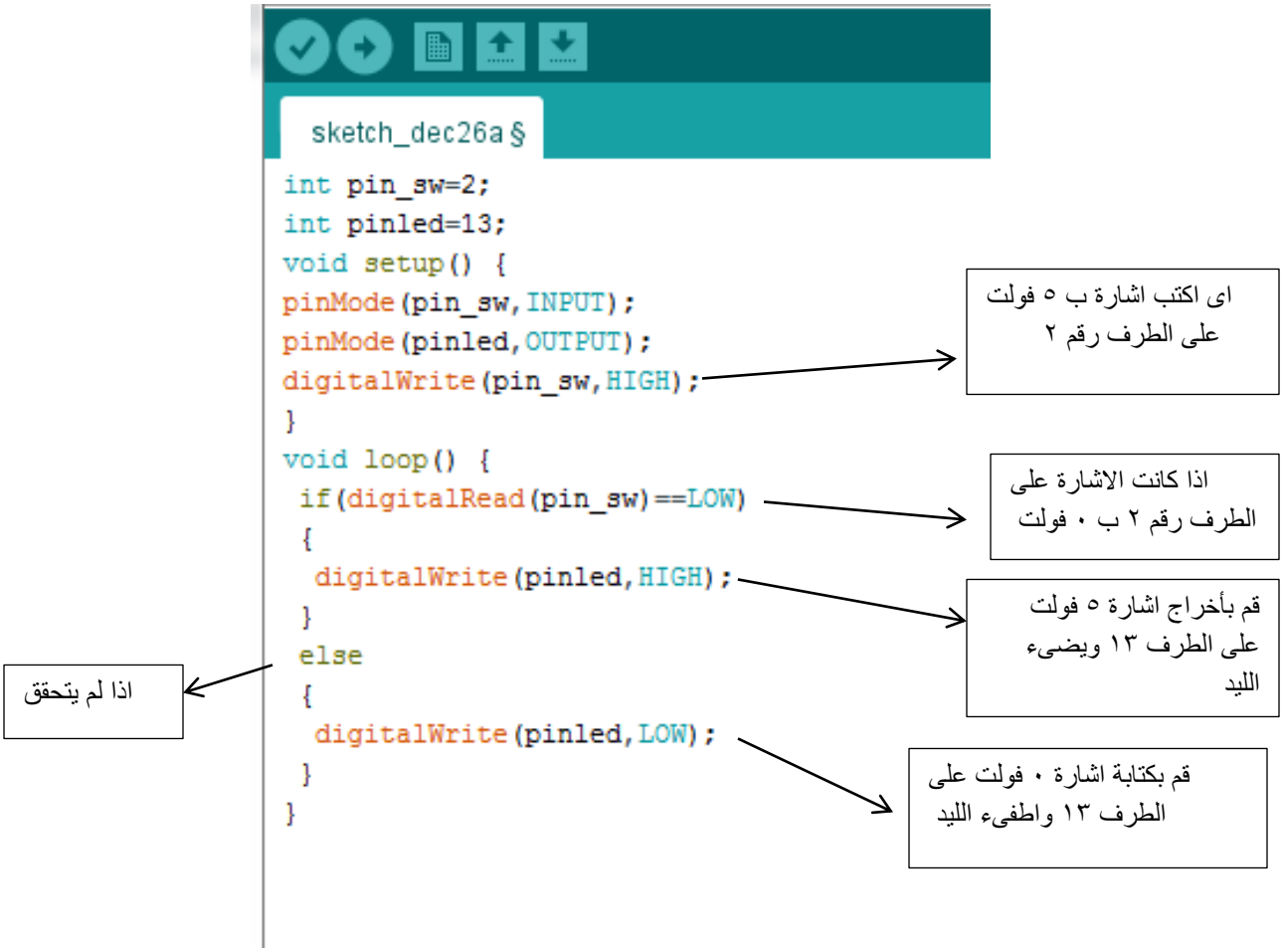




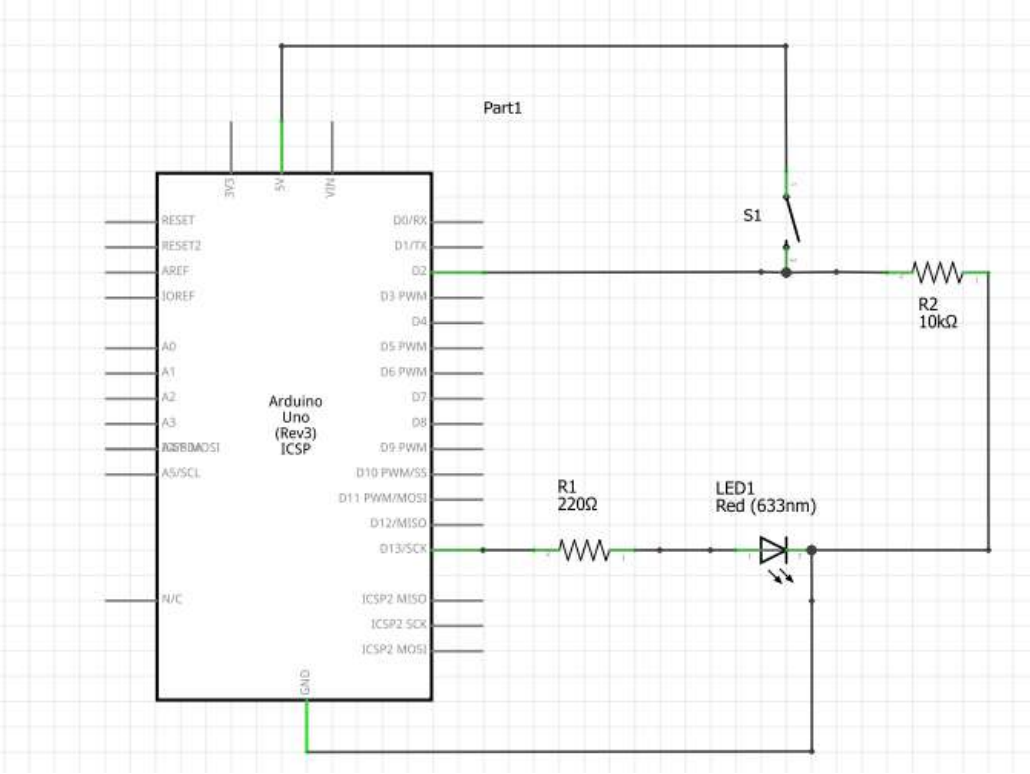
المكونات الالكترونية :

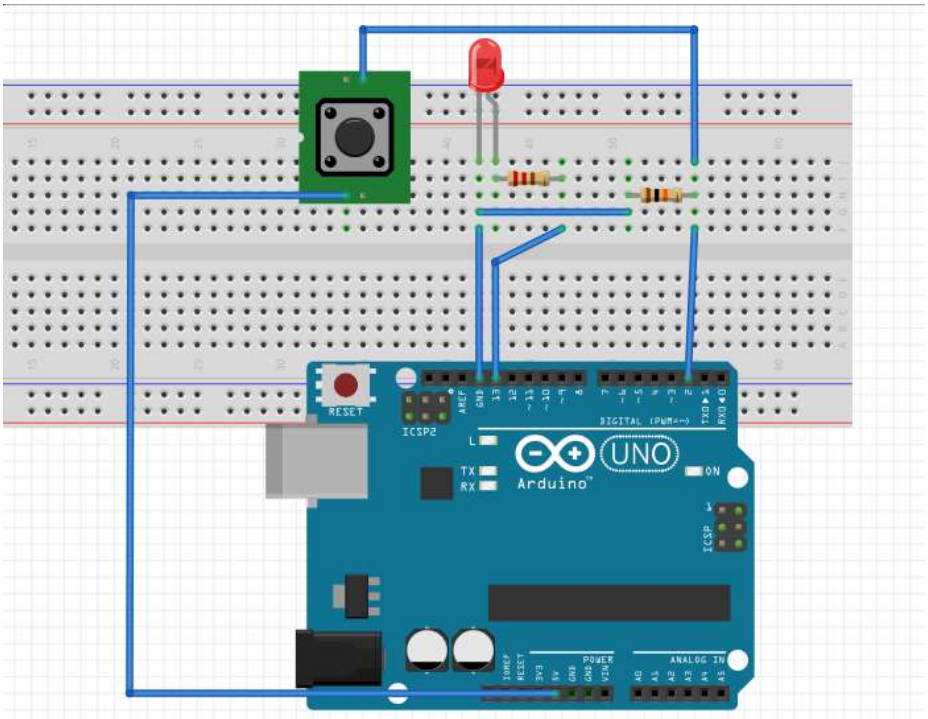
لوحة اختبار – ليد – اردوينو اونو – مجموعة اسلاك للتوصيل – مقاومة ثابتة ٢٢٠ اوم او ٣٢٠ اوم – مفتاح push buttons PCB

كود البرنامج :



يمكن تنفيذ المشروع السابق ولكن مع تغيير التوصيل وجزء من الكود كما يلى :


Part1
S1
R2
10kΩ
R1
220Ω
LED1
Red (633nm)
Arduino
Uno
(Rev3)
ICSP
RESET
RESET2
AREF
IOREF
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
ICSP2 MISO
ICSP2 SCK
ICSP2 MOSI
GND


RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN

كود البرنامج :

ماهى جملة IF الشرطية :

هى جملة تكتب للتحقق من شرطا ما فأذا تحقق يتم تنفيذ الحدث الاول واذا لم يتحقق لايتم تنفيذ شىء

وتأخذ جملة if الشرطية الصيغة التالية

If (الشرط المراد اختباره)

{

الحدث المراد تنفيذه فى حالة تحقق الشرط

}

مثال : طباعة الاعداد من ٠ الى ٥

```
If (integer > 5)
{
Print (i)
}
```

نوع اخر من جملة if الشرطية وهو

جملة if else

وتأخذ الصيغة التالية

```
If (الشرط المراد اختباره)
{
نفذ الحدث الاول اذا تحقق الشرط
}
Else
{
نفذ الحدث الثانى اذا لم يتحقق الشرط
}
```

هناك نوع اخر من جملة if الشرطية وهو

جملة if else المتداخلة

وفيها يتم اختبار عدد معين من الشروط

وتأخذ الصيغة الاتية

```
If ( الشرط الاول المراد اختباره)
{
 نفذ الحدث الاول اذا تحقق الشرط الاول
}
Else if ( الشرط الثانى المراد اختباره )
{
 نفذ الحدث الثانى اذا تحقق الشرط الثانى
}
Else
{
 نفذ الحدث الثالث اذا لم يتحقق الشرط الاول والثانى
}
```

معاملات المقارنة المستخدمة فى عملية اختبار الشرط :

X==y

اى ان x تساوى y

X!=y

اى ان x لا تساوى y

X<y

اى ان x اصغر من y

x>y

اى ان x اكبر من y

x<=y

اى ان x اصغر من او يساوى y

x>=y

اى ان x اكبر من او تساوى y

معاملات بولين المستخدمة مع جملة if الشرطية :

&& (logic and)

وتمثل العلامة && اى انه لكى يتم تحقيق X لابد من تحقيق Y1 and Y2

وتكتب بالصورة الاتية X= Y1 && Y2

الصيغة عند استخدمها مع جملة if

```
If (a> 10 && b>20)
{
}
```

|| (logic or)

وتمثل العلامة || اى انه لكى يتم تحقيق X لابد من تحقيق Y1 او Y2

وتكتب بالصورة الاتية X= Y1 || Y2

الصيغة عند استخدمها مع جملة if

```
If (a > 0 || y > 0)
{
}
```

! (not)

وتمثل العلامة ! اى انه نفى قيمة المتغير بمعنى اذا كانت x=0 تكون النتيجة !x=1

الصيغة عند استخدمها مع جملة if

```
If ( ! X)
{
}
```

جملة switch case :

وهى تشبة جملة if المتدخلة وتأخذ الشكل التالى :

```
Switch ()
{
Case 1:
Break;
 Case 2:
Break;
Default:
Break;
}
```

مثال :

```
switch (input) {
case 'm':
print(“hello mohammed”);
break;
case 'a':
print(“hello ahmed”);
break;
case 'c':
print(“hello cella”);
break;
case 'e':
print(“hello eman”);
break;
default:
```

فى حالة اذا كانت input تساوى m فاطبع hello mohammed

اخرج من الاختبار

فى حالة اذا كانت input تساوى a فاطبع hello ahmed

البرنامج (٢)

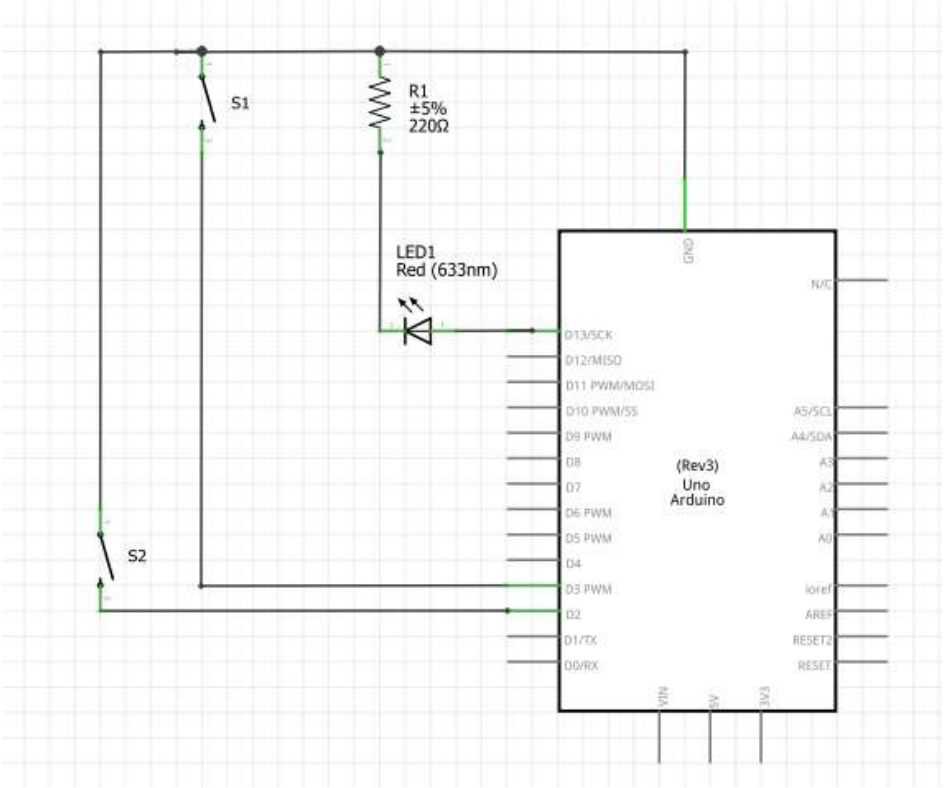
دائرة اردوينو تحتوى على مفتاحين وليد

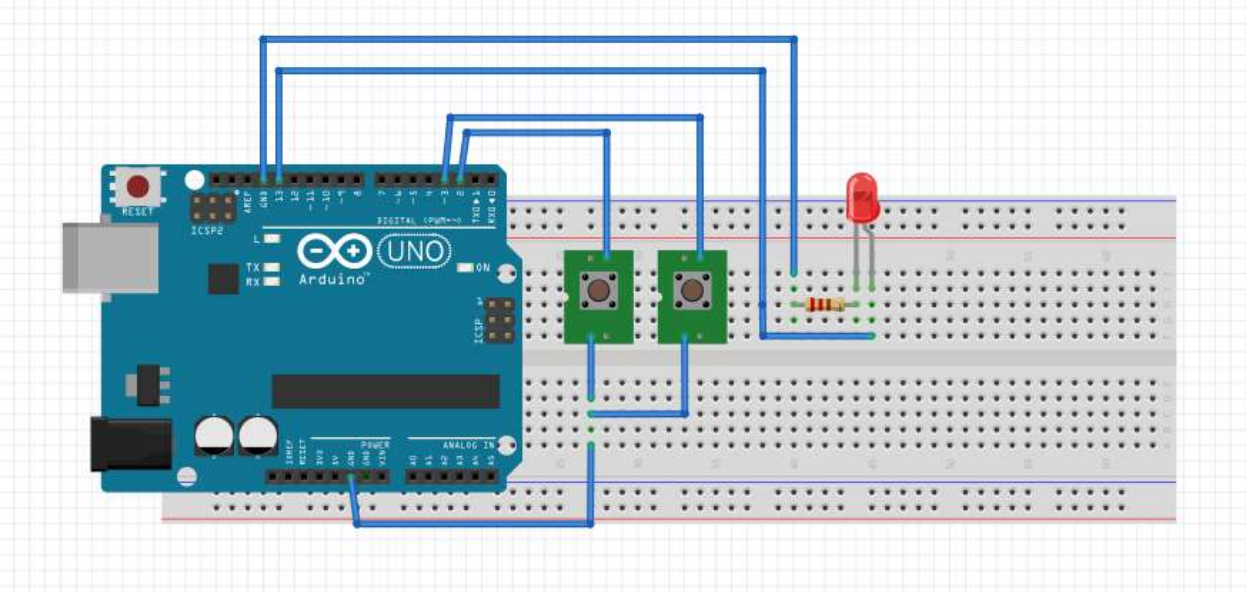
المفتاح الاول عند الضغط عليه يضىء الليد بينما عند الضغط على المفتاح الثانى يطفىء الليد

المكونات الالكترونية :

لوحة اختبار – ليد – اردوينو اونو – مجموعة اسلاك للتوصيل – مقاومة ثابتة ٢٢٠ اوم او ٣٢٠ اوم – مفتاحين push buttons PCB

دائرة التوصيل :





كود البرنامج :

sketch_jan02a §

```
int pin_sw1=2;
int pin_sw2=3;
int pinled=13;

void setup() {
pinMode(pin_sw1,INPUT);
pinMode(pin_sw2,INPUT);
pinMode(pinled,OUTPUT);
digitalWrite(pin_sw1,HIGH);
digitalWrite(pin_sw2,HIGH);
}
void loop() {
if(digitalRead(pin_sw1)==LOW)
{
  digitalWrite(pinled,HIGH);
}
else if(digitalRead(pin_sw2)==LOW)
{
  digitalWrite(pinled,LOW);
}
}
```

تعريف المتغيرات بأرقام اطراف الاردوينو

تخصيص المخارج و المداخل

كتابة اشارة ٥ فولت على الاطراف المخصصة

اختبار اذا كانت pin_sw1 تساوى زيرو فولت

اكتب على pinled اشارة ٥ فولت

اختبار اذا كانت pin_sw2 تساوى زيرو فولت

اكتب على pinled اشارة زيرو فولت

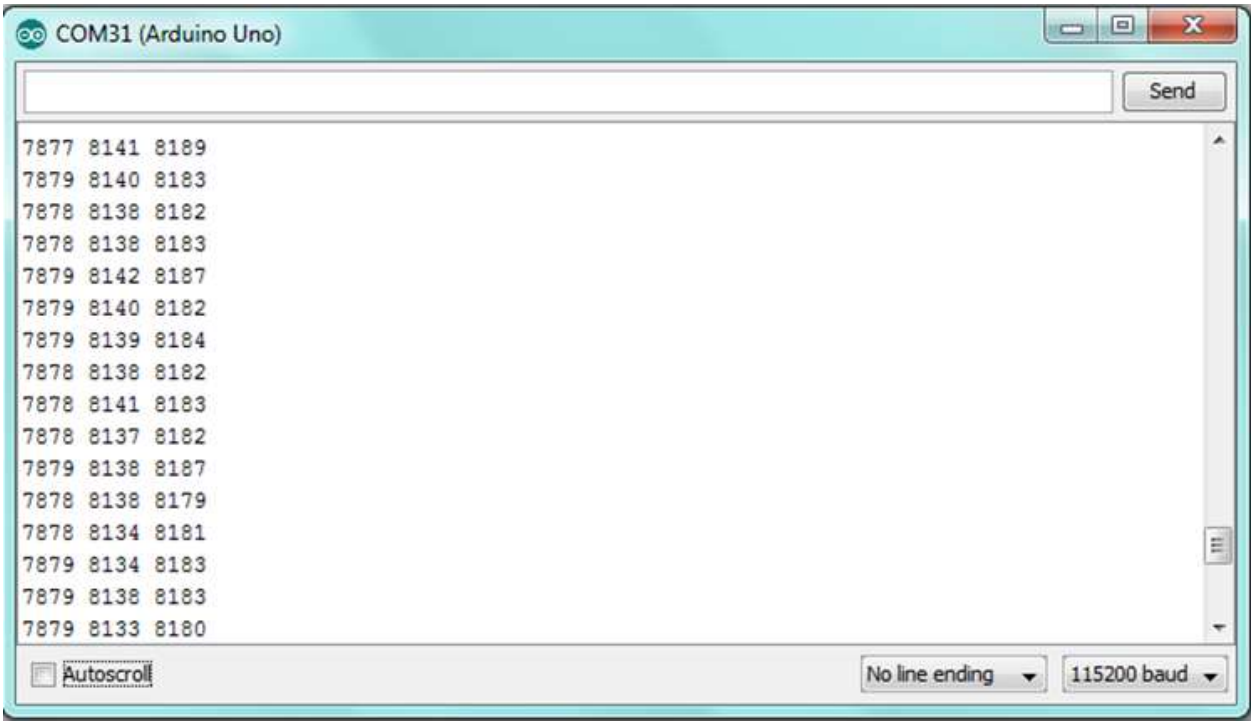
Done compiling.

كما يمكن تعديل وظيفة البرنامج السابق ليعمل بهذه الطريقة

فعند الضغط على المفتاح الاول يضيء الليد بعد مرور ثانيتين اما عند الضغط على المفتاح الثانى يطفيء الليد بعد مرور ثانيتين

المراقب التسلسلى : serial monitor

وهى عبارة عن نافذة فى برنامج IDE arduino تعمل على ارسال واستقبال البيانات من الكمبيوتر الى الاردوينو .

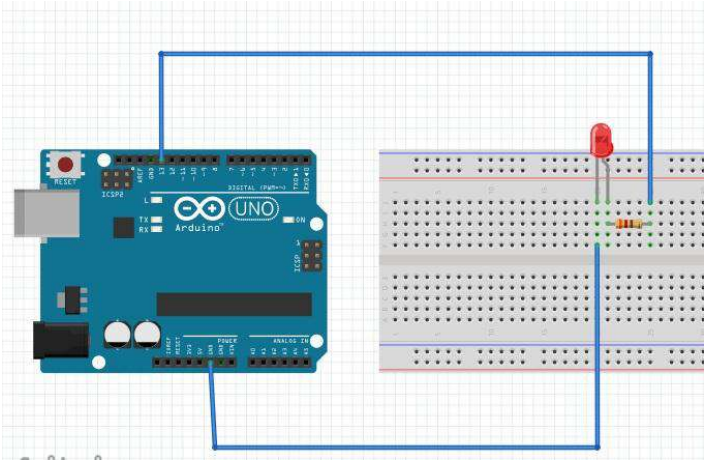


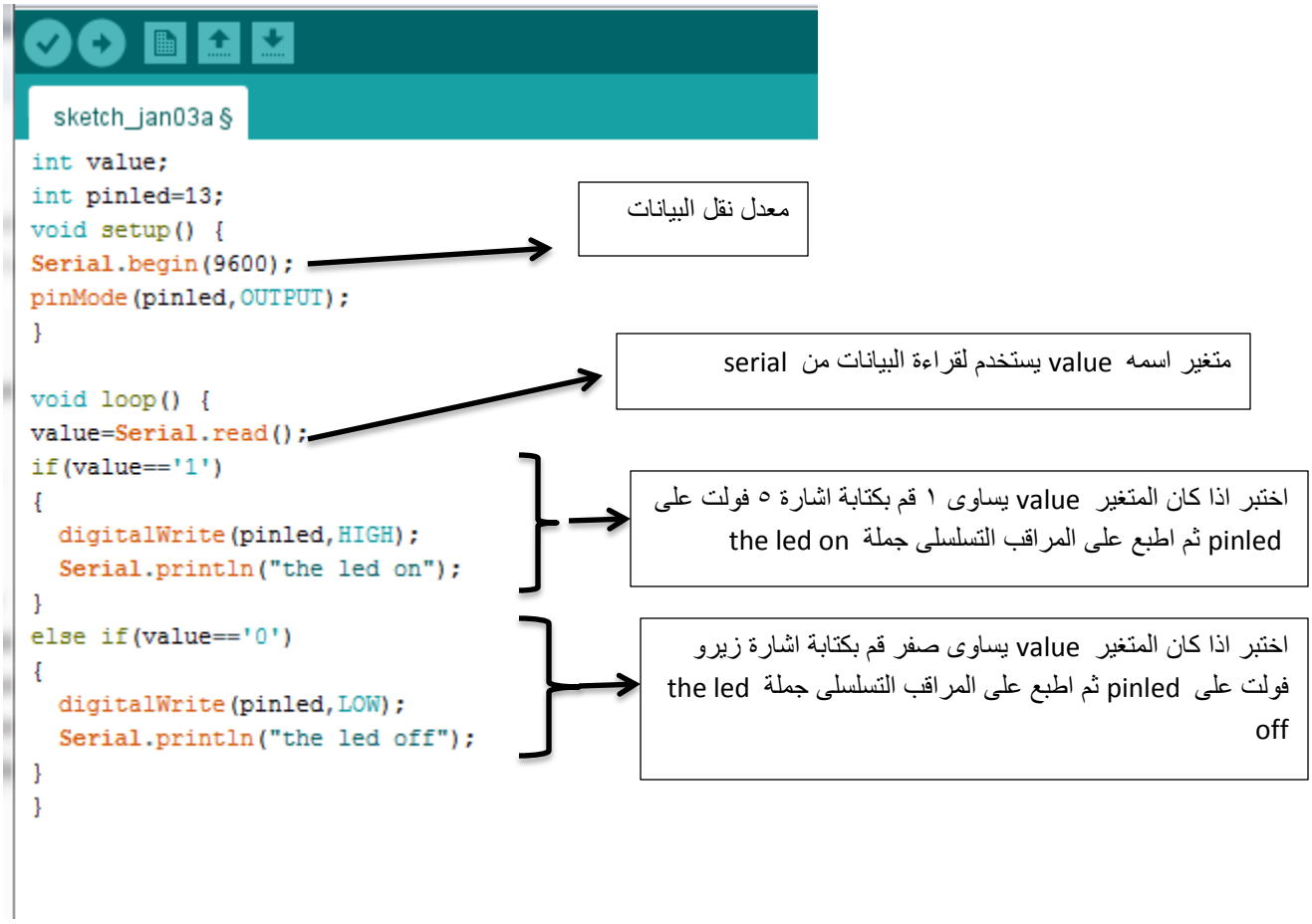
والامثلة التالية توضح فهم واستخدام serial monitor

البرنامج (١)

التحكم فى تشغيل واطفاء ليد عن طريق المراقب التسلسلى .

دائرة التوصيل :





كيف يعمل البرنامج ؟

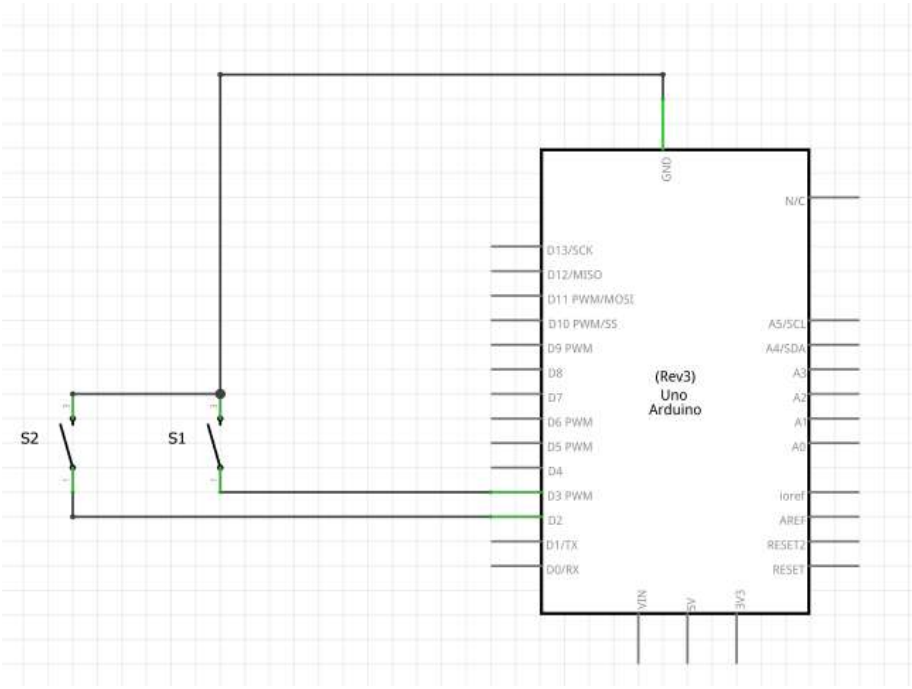
بعد كتابة الكود يتم الضغط على مفتاح المراقب التسلسلى والذى يشبه العدسة كما بالشكل التالى

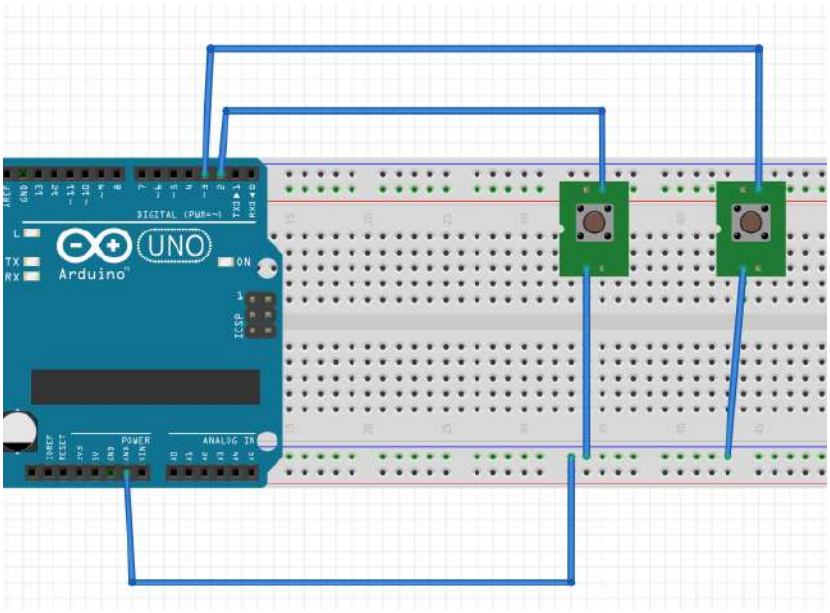
فتظهر لنا نافذة المراقب التسلسلى قم بكتابة العدد ١ ونضغط على مفتاح send سنلاحظ اضاءة الليد واذا تم كتابة العدد ٠ ثم الضغط على مفتاح send فسنلاحظ اطفاء الليد

البرنامج (٢)

تتكون الدائرة من مفتاحين فقط فعند الضغط على اى مفتاح منهم يتم كتابة اسم المفتاح الذى تم الضغط عليه فى نافذة المراقب التسلسلى .

دائرة التوصيل :





كود البرنامج :

```
int pin_sw1=2;
int pin_sw2=3;

void setup()
{
  pinMode(pin_sw1,INPUT);
  pinMode(pin_sw2,INPUT);
  digitalWrite(pin_sw1,HIGH);
  digitalWrite(pin_sw2,HIGH);
  Serial.begin(9600);
}
void loop()
{

  if(digitalRead(pin_sw1)==LOW)
  {
    delay(500);
   Serial.println("press switch number 1");

  }
  else if (digitalRead(pin_sw2)==LOW)
  {
    delay(500);
   Serial.println("press switch number 2");
  }

}
```

Done compiling.

اسم المفتاح الاول الذى سيتم طباعتة فى المراقب التسلسلى عند الضغط عليه

اسم المفتاح الثانى الذى سيتم طباعتة فى المراقب التسلسلى عند الضغط عليه

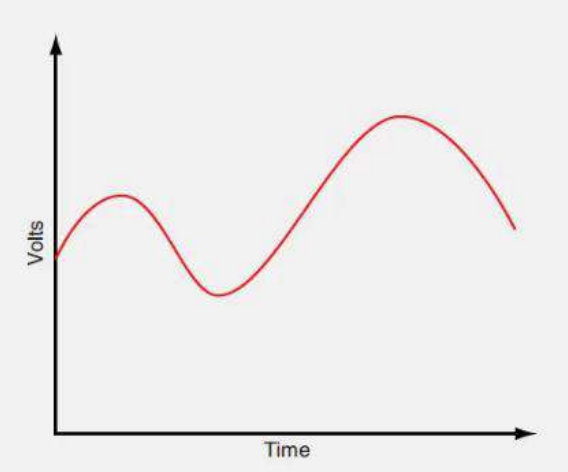
التعامل مع المدخلات التناظرية فى الاردوينو





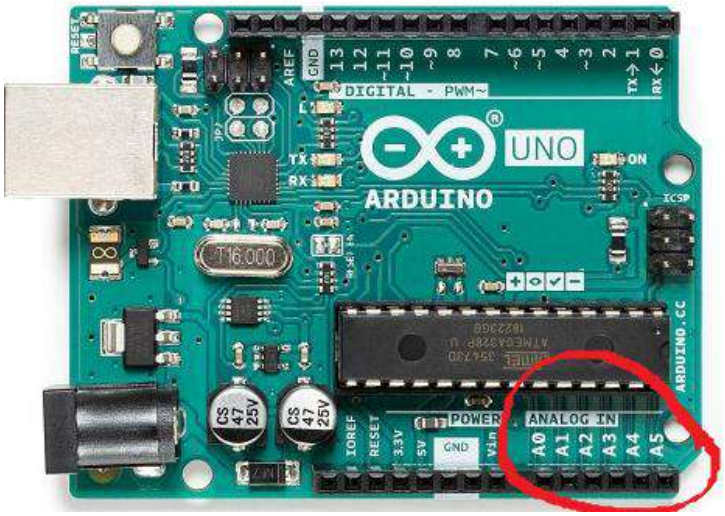

المدخلات التناظرية analog input :

وهى عبارة عن قيمة سواء جهد او تيار وتكون هذه القيمة متغيرة بالنسبة للزمن حيث ان كل قيمة تناظرية تمثل امامها قيمة مختلفة عن الاخرى بالنسبة للزمن .



حيث تحتوى لوحة الاردوينو على ٦ قنوات يمكنها التحويل من القيمة التناظرية الى القيمة الرقمية و اطراف ٦ قنوات على الاردوينو هى :

A0 – A1 - A2 – A3 – A4 – A5



دقة التحويل من القيمة التناظرية الى القيمة الرقمية فى الاردوينو تبلغ ١٠ بت وتأخذ القيمة من ٠ الى ١٠٢٣ وللتوضيح اكثر نذكر المثال التالى :

ولنفرض ان هناك جهد يتغير من ٠ الى ٥ فولت على الطرف A0 فسوف يتم عمل map على الجهد من ٠ الى ٥ فولت يمكن تمثيلة ما بين القيم الصحيحة من ٠ الى ١٠٢٣

ولقراءة قيمة تناظرية من الاردوينو نستخدم الكود التالى :

Analog Read ( )

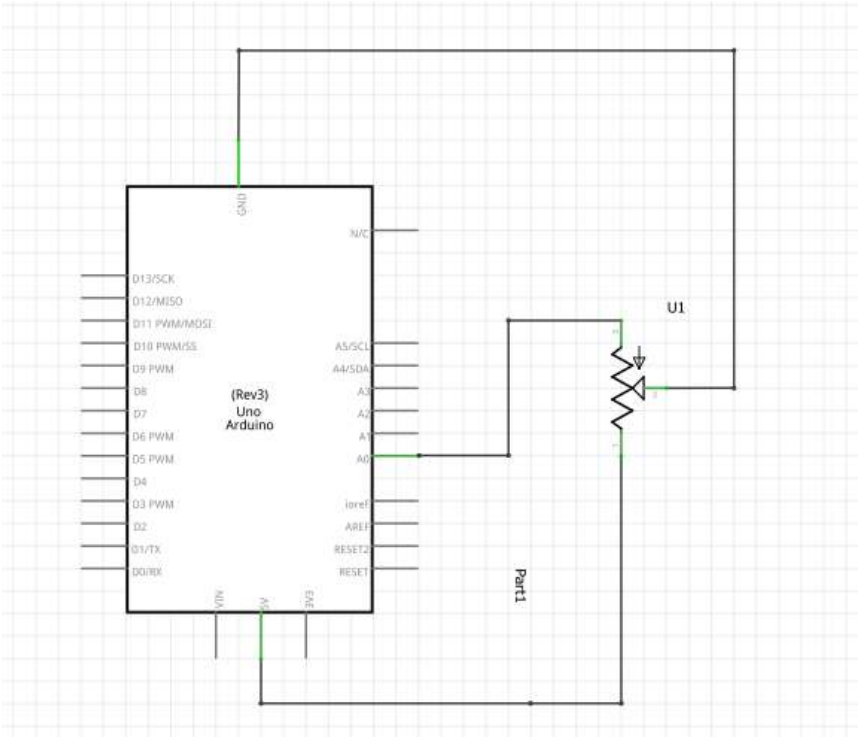
البرنامج (١)

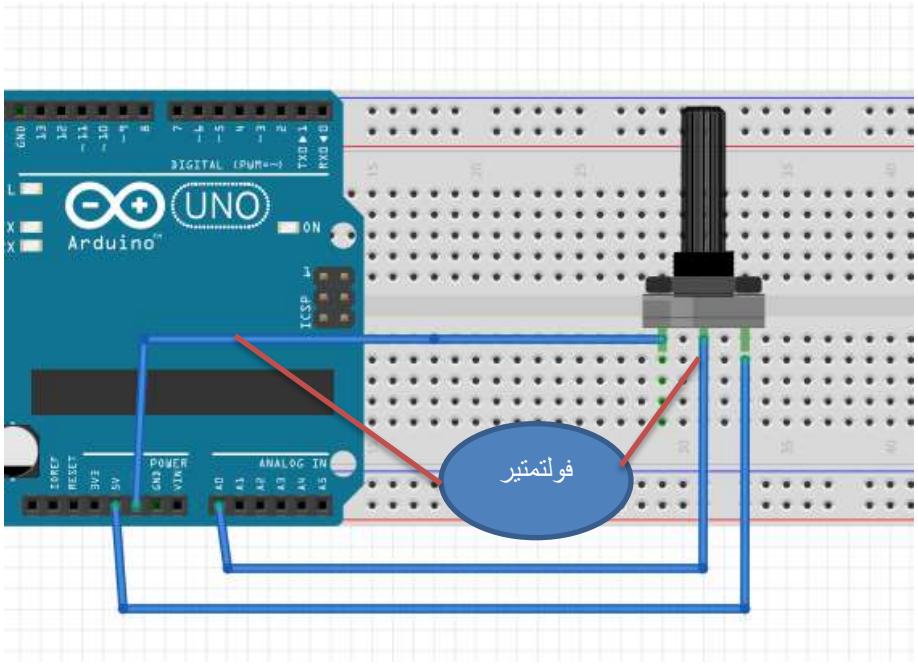
نريد ان نجد قيمة التحويل بأستخدام مقاومة متغيرة قيمتها 2.2k ohm وعرضها على المراقب التسلسلى ومقارنتها بجهاز قياس الجهد (الفولتميتر)

مكونات الدائرة :

اردوينو اونو – اسلاك توصيل – مقاومة متغيرة قيمة ٢,٢ كيلو اوم او ١٠ كيلو اوم – لوحة اختبار

دائرة التوصيل :





كود البرنامج :

```
int analogPin = 0;

int val = 0;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  val = analogRead(analogPin);
  Serial.println(val);
  delay(2000);
}
```

متغير val يقرأ قيمة تناظرية من الطرف A0

اطبع قيمة المتغير val فى المراقب التسلسلى

تأخير مابين كل قراءة والاخرى مدة ثانيتين

Done compiling.

بعد تحميل الكود على الاردوينو نقوم بفتح نافذة المراقب التسلسلى ونبدأ بتغير قيمة المقاومة الى ان تصل قيمة المحول فى المراقب التسلسلى الى القيمة ٢٠٥

بعد ذلك نقوم بالعملية الحسابية البسيطة الاتية :

حيث ان اعلى قيمة للمحول فى حالة ال ٥ فولت تساوى ١٠٢٣

والمطلوب هنا معرفة الجهد المناظر للقيمة ٢٠٥

وبالتالى فان الجهد الموجود على اطراف المقاومة المتغيرة تساوى ٥ فولت مضروبة فى ٢٠٥ ثم نقسم الناتج على ١٠٢٣

من العملية السابقة يكون الناتج هو ١ فولت وهى نفس القيمة التى يقرأها الافوميتر على اطراف المقاومة .

ومن ذلك نستنتج ان ١ فولت يناظر القيمة ٢٠٥ من اصل القيمة الكلية ١٠٢٣

البرنامج (٢)

عرض قيمة الجهد على اطراف المقاومة المتغيرة من خلال المراقب التسلسلى

دائرة التوصيل نفس السابقة

```
int analogPin = 0;
    int volt_restance;
int val = 0;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  val = analogRead(analogPin);
   volt_restance=(5*val)/1023;
   Serial.println("volt=");
  Serial.println(volt_restance);
  delay(2000);
}
```

للتحويل من قيمة المحول الى قيمة تناظر الجهد فأننا نقوم بضرب اعلى قيمة للجهد وهى ٥ فولت مع قيمة المحول الحالية والناتج نقسمة على اعلى قيمة للمحول والنتيجة الكلية نضعها فى المتغير volt_restance

Done compiling.

الحساسات التناظرية analog sensor :

اى حساس تناظرى يحتوى على ثلاث اطراف اساسية وهما :

الاول : تغذية الجهد

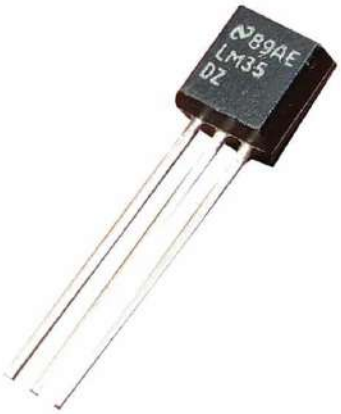
الثانى : الارضى

الثالث : خرج اشارة الحساس وتكون قيمتها متغيرة

اما لو وجدت اربع اطراف فستكون للاشارة الرقمية digital output

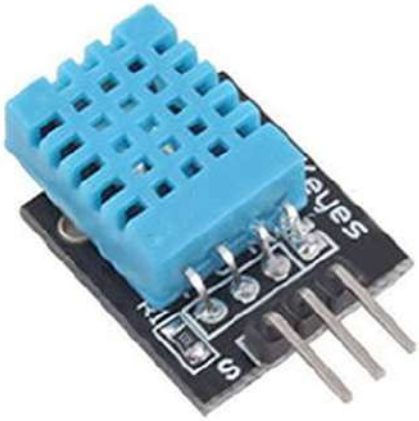
ومن انواع الحساسات التناظرية المستخدمة فى الاردوينو :

حساس قياس درجة الحرارة LM 35

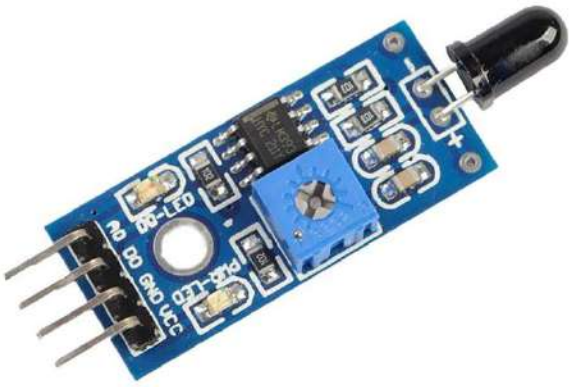


حساس قياس غاز الكحول MQ3

حساس قياس الرطوبة



حساس قياس اللهب :



حساس قياس الاحمال او الاوزان :

حساس قياس الغاز MQ5 :

حساس قياس غاز الهيدروجين MQ8 :

حساس قياس المسافات :

البرنامج (١)

قياس درجة الحرارة بأستخدام الحساس LM35 وعرض قيم درجات الحرارة فى المراقب التسلسلى

اولا : مخطط اطراف الحساس

ثانيا : خطوات ايجاد درجة الحرارة

سنقوم بتحويل قيمة خرج LM35 الى مللى فولت عن طريق المعادلة التالية :

Value mv = (read value from sensor / 1023)*5000

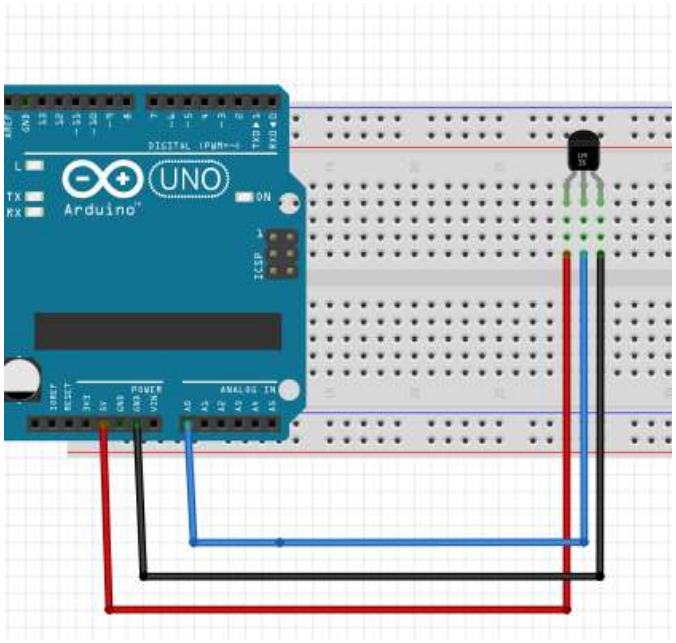
من المعروف ان قيمة الخرج للحساس تمثل ان لكل ١٠ مللى فولت تمثل درجة واحدة سيليزس فتمثل بالمعادلة الاتية

Celsius = Value mv / 10

وللتحويل من سيليزس الى فرنهيت نستخدم العلاقة التالية

F°=(C°×1, 8) +32

مخطط التوصيل :



كود البرنامج :

```
int outputpin= 0;
void setup()
{
Serial.begin(9600);
}


void loop()
{
int rawvoltage= analogRead(outputpin);
float millivolts= (rawvoltage/1023.0) * 5000;
float celsius= millivolts/10;

float Fahrenheit=(celsius*9/5)+32;
Serial.println(" degrees Celsius, ");
Serial.println(celsius);
Serial.println(" Fahrenheit ");
Serial.println(Fahrenheit);
delay(2000);

}
```

المعادلة التى تم شرحها وتستخدم لتحويل قيمة المحول الى جهد قيمته مللى فولت

تحويل قيمة الجهد الى درجة حرارة سيليزس

تحويل درجة الحرارة سيليزس الى فرنهيت

Done compiling.

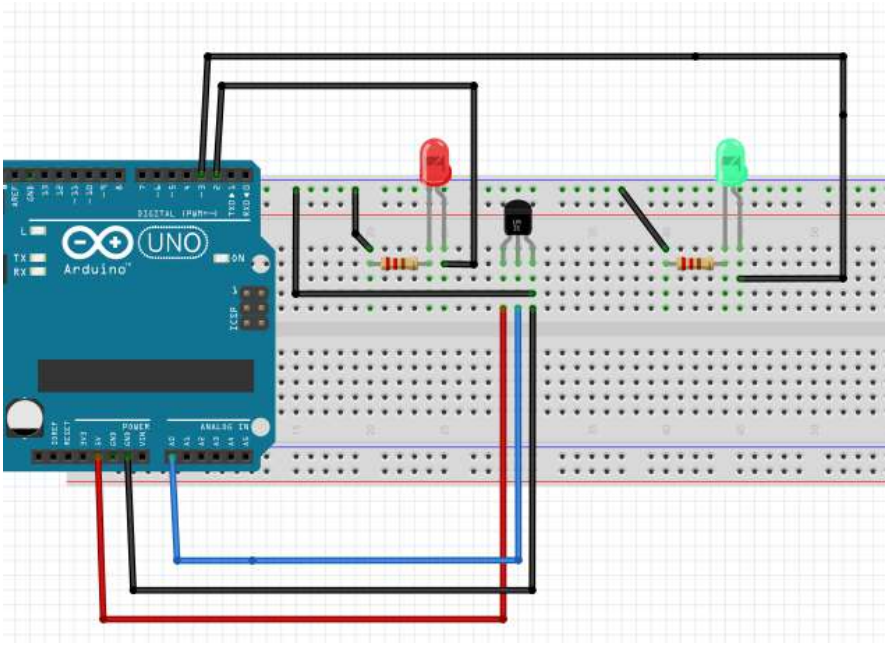
البرنامج (٢)

قياس درجة الحرارة وعرضها فى المراقب التسلسلى مع ملاحظة انه عندما تكون درجة الحرارة اقل من او تساوى ٣٠ درجة يتم اضاءة الليد الاخضر اما اذا كانت درجة الحرارة اقل من او تساوى ٣٢ يتم اضاءة الليد الاحمر

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – حساس حرارة LM 35 – مقاومات ٢٢٠ اوم – اردوينو اونو – ليد اخضر – ليد احمر

مخطط التوصيل :



كود البرنامج :

```
int outputpin= 0;
int pinled1=2;
int pinled2=3;
void setup()
{
 pinMode(pinled1,OUTPUT);
 pinMode(pinled2,OUTPUT);
Serial.begin(9600);
}
void loop(){
int rawvoltage= analogRead(outputpin);
float millivolts= (rawvoltage/1023.0) * 5000;
float celsius= millivolts/10;
Serial.println(" degrees Celsius, ");
Serial.println(celsius);
if(celsius<=30)
{
 digitalWrite(pinled1,HIGH);
digitalWrite(pinled2,LOW);
}
else if(celsius<=32)
{
 digitalWrite(pinled2,HIGH);
digitalWrite(pinled1,LOW);
}
delay(2000);
}
```

Done compiling.

المقاومة الضوئية photo resistor

ويطلق عليها LDR وهى اختصار لجملة

Light dependent resistor

وهى مقاومة تعتمد على شدة الاستضاءة حيث تتوقف قيمة المقاومة على شدة الضوء فكلما زادت شدة الاضاءة تقل المقاومة وكلما قلت الاضاءة تزيد المقاومة

البرنامج (٣)

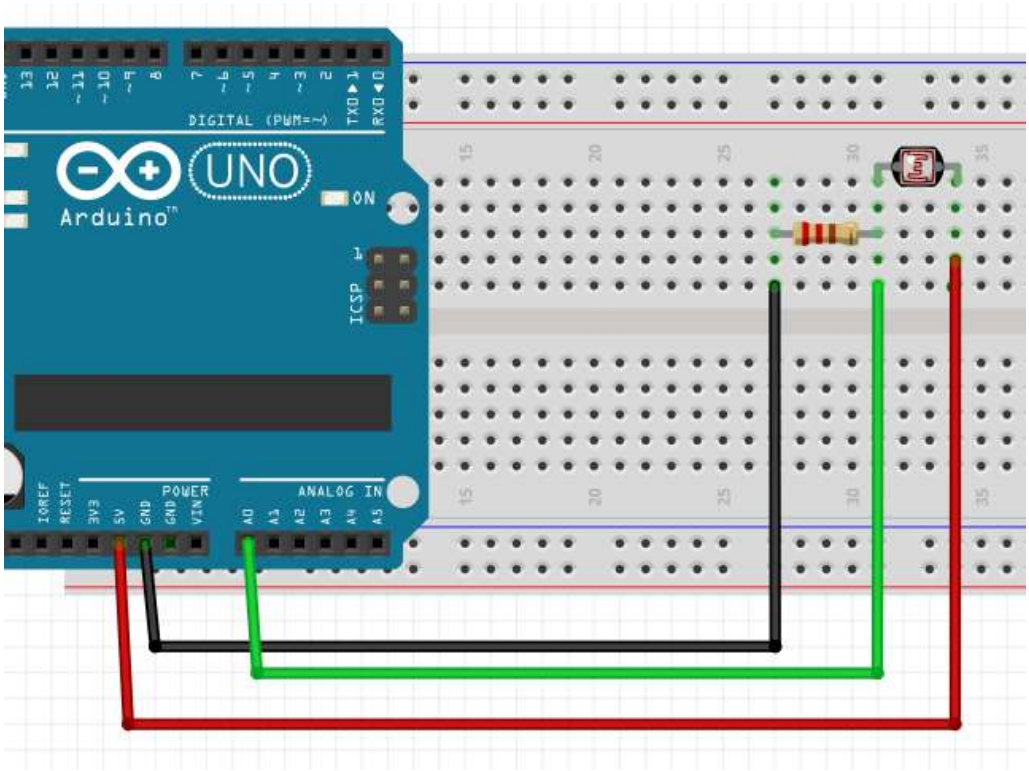
طباعة القيم التى تقرأها المقاومة الضوئية فى المراقب التسلسلى

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – مقاومة ١٠ كيلو اوم – اردوينو اونو – مقاومة ضوئية

دائرة التوصيل :

قيمة المقاومة المتصلة مع LDR هى ١٠ كيلو اوم



كود البرنامج :

```
int photocell=A0;
int photocell_reading;
void setup()
{
 Serial.begin(9600);
}
void loop()
{
 photocell_reading=analogRead(photocell);
Serial.println("light value=");
Serial.println(photocell_reading);
delay(2000);
}
```

Done compiling.

البرنامج (٤)

استشعار الضوء والظلام ففى حالة وجود ضوء يكتب الاردوينو فى المراقب التسلسلى light وفى حالة انعدام او انخفاض الضوء يتم طباعة dark .

دائرة التوصيل : نفس الدائرة السابقة

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – مقاومة ١٠ كيلو اوم – اردوينو اونو – مقاومة ضوئية

```
int photocell=A0;
int photocell_reading;
void setup()
{
 Serial.begin(9600);
}
void loop()
{
 photocell_reading=analogRead(photocell);
Serial.println("light value=");
Serial.println(photocell_reading);
delay(2000);
if(photocell_reading<=120)
{
  Serial.println("dark");
}
else if(photocell_reading >=200)
{
  Serial.println("light");
}
delay(1000);
}
```

ولكن من اين حصلنا على القيم ١٢٠ و ٢٢٠ التى سوف نختبر الشرط

والاجابة حيث يتم الحصول على هذه القيم من البرنامج السابق والذى يتم منه قراءة اعلى واقل قيمة للضوء

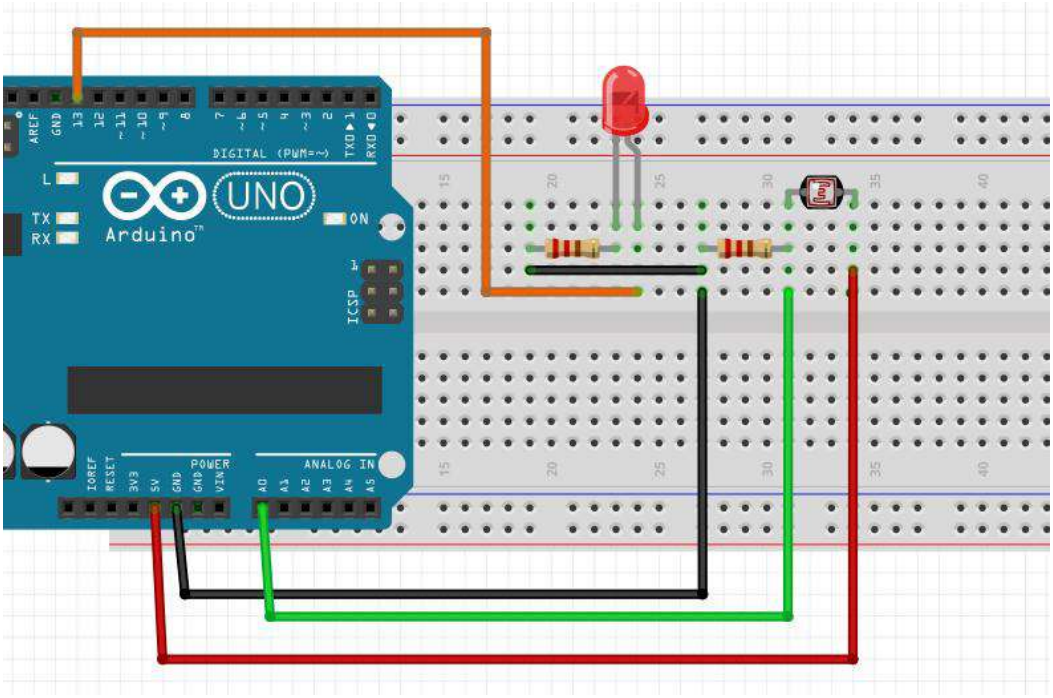
البرنامج (٥)

استشعار الضوء والظلام ففى حالة وجود ضوء يكتب الاردوينو فى المراقب التسلسلى light مع اضاءة الليد الاحمر وفى حالة انعدام الضوء يتم طباعة dark مع اطفاء الليد الاحمر

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – مقاومة ١٠ كيلو اوم – اردوينو اونو – مقاومة ضوئية – ليد – مقاومة ٢٢٠ اوم

دائرة التوصيل :



كود البرنامج :

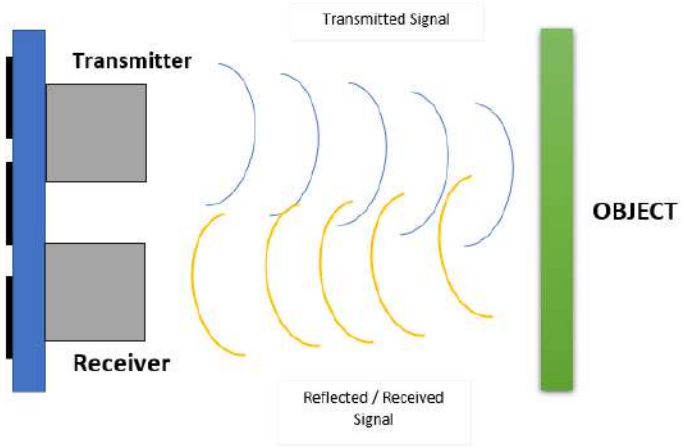
```
int photocell=A0;
int photocell_reading;
int pinled=13;
void setup()
{
 Serial.begin(9600);
 pinMode(pinled,OUTPUT);
}
void loop()
{
 photocell_reading=analogRead(photocell);
Serial.println("light value=");
Serial.println(photocell_reading);
delay(2000);
if(photocell_reading<=120)
{
  Serial.println("dark");
  digitalWrite(pinled,HIGH);
  delay(1000);
}
else
{
  digitalWrite(pinled,LOW);
  delay(1000);
}
}
```

Done compiling.

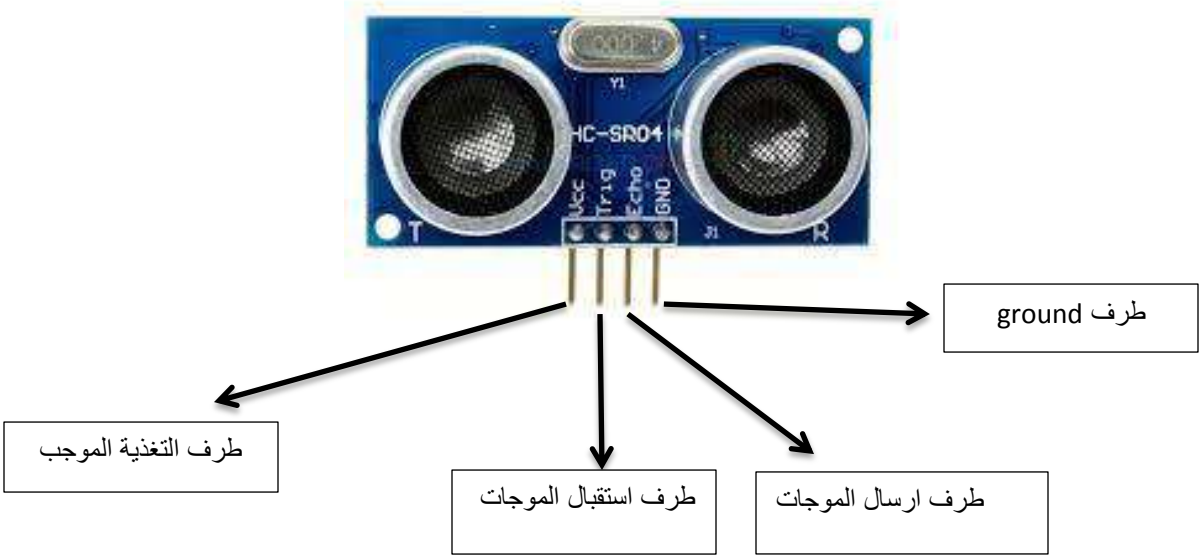
حساسات قياس المسافات او الحساسات الفوق صوتية

Ultrasonic sensor module HC – SR04

يعتمد مبدأ عمل هذه الحساسات على الارسال والاستقبال . حيث يتم ارسال موجات فوق صوتية على الجسم المراد تحديد مسافتة ثم تنعكس هذه الموجات مرة اخرى فيتم استقبالها وعلى اساس حساب الفترة الزمنية فى الارسال والاستقبال يتم تحديد المسافة



شرح اطراف حساس الموجات الفوق صوتية



حيث ان trigger اى استقبال صدى الموجات الفوق الصوتية اما Echo فهو ارسال الموجات الفوق صوتية

اقصى مدى يمكن قياسه هو ٤ متر واقل مدى يمكن قياسه ٢ سم

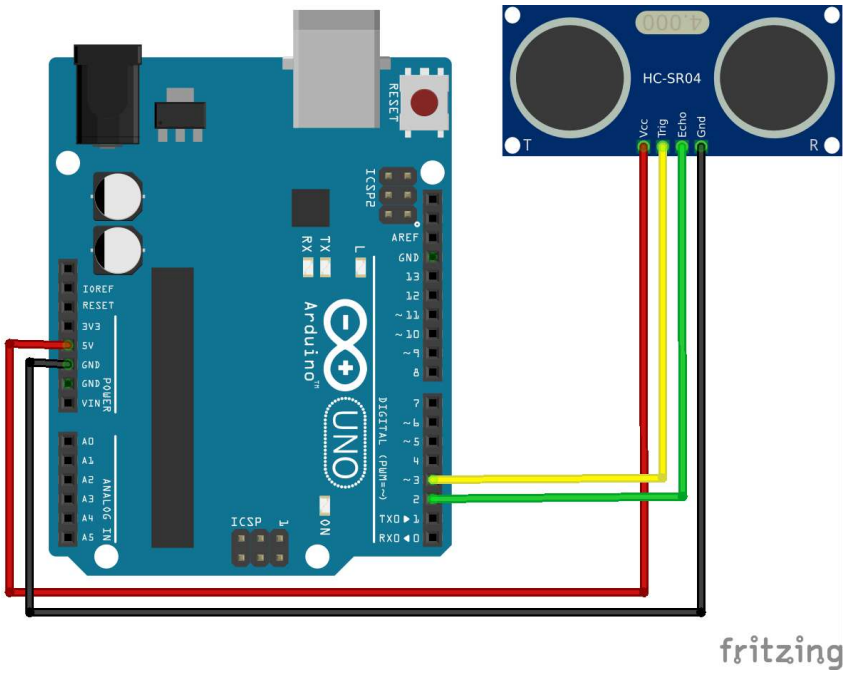
البرنامج (٦)

حساب المسافة بين الحساس والاجسام وطباعة المسافة فى المراقب التسلسلى بأستخدام حساس الموجات الفوق الصوتية

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – حساس الموجات الفوق صوتية – اردوينو اونو

دائرة التوصيل :



كود البرنامج :

```
 int trigPin =3;
 int echoPin = 2;
 long duration;
 long  distance;
void setup()  {
  Serial.begin(9600);
 pinMode(trigPin,OUTPUT);
 pinMode(echoPin,INPUT);
}
void loop() {
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);
  duration=pulseIn(echoPin,HIGH);
  distance=duration/58;
  delay(100);
  Serial.print(distance);
  Serial.println(" cm" );
}
```

Done compiling.

نلاحظ من الكود السابق انه تم قسمة duration على ٥٨ حيث انه تم ذكر فى datasheet الخاصه بالحساس المعادلات الاتية :

فى حالة الحصول على المسافة بالسنتيمتر يتم القسمة على ٥٨ اما فى حالة الحصول على المسافة بالبوصة يتم القسمة على ١٤٨

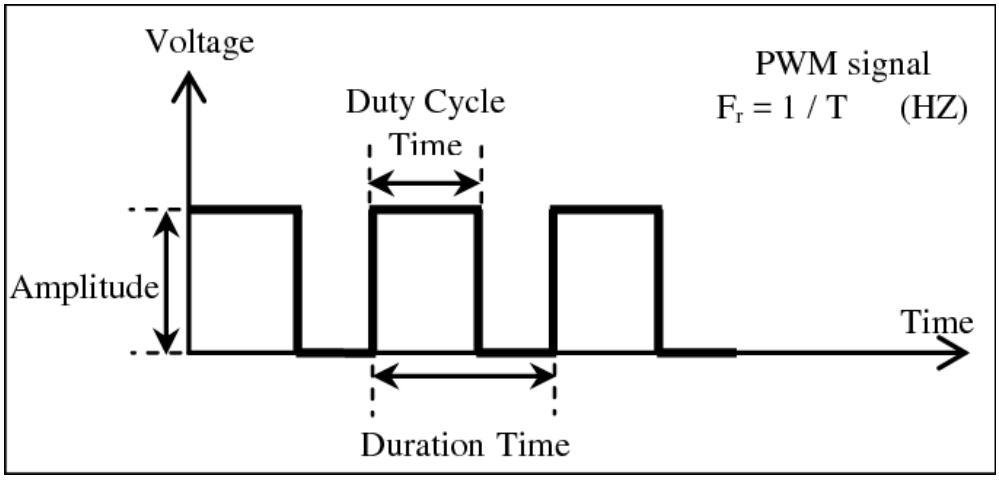
تعديل عرض النبضة :

Pulse width modulation (PWM)

وتستخدم فى الحصول على عدة جهود مختلفة من جهد واحد

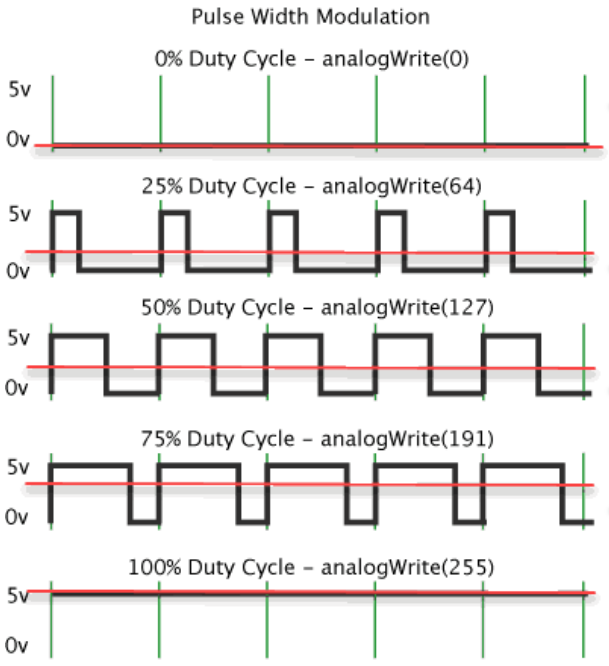
حيث يمكن الحصول على مدى من الجهود مابين ال ٠ الى ٥ فولت

تمثل PWM موجة مربعة او مستطيلة وتأخذ الشكل التالى :



ماهو duty cycle ؟

وهو النسبة بين الموجة فى حالة on والموجة فى حالة off



فى حالة تعديل عرض النبضة بصفر نحصل على جهد زيرو فولت

فى حالة تعديل عرض النبضة بنسبة ٢٥% من نسبة جهد الدخل فأننا نحصل على جهد ١,٢٥ فولت

فى حالة تعديل عرض النبضة بنسبة ٥٠% من نسبة جهد الدخل فأننا نحصل على جهد ٢,٥ فولت

فى حالة تعديل عرض النبضة بنسبة ٧٥% من نسبة جهد الدخل فأننا نحصل على جهد ٣,٧٥ فولت

فى حالة تعديل عرض النبضة بنسبة ١٠٠% من نسبة جهد الدخل فأننا نحصل على جهد ٥ فولت

فعن طريق duty cycle يتم التحكم فى الجهد فمثلا

Vout =0/100 x 5 = 0

Vout=25/100 x 5 = 1.25 v

Vout=50/100 x 5 = 2.5 v

Vout=100/100 x 5 = 5 v

فبما انه يمكن التحكم فى الجهد الخارج بواسطة التحكم فى عرض النبضة فيمكن استخدام هذه الخاصية فى التحكم بشدة الاضاءة والتحكم بسرعة المحركات حيث ان :

التردد فى PWM يساوى

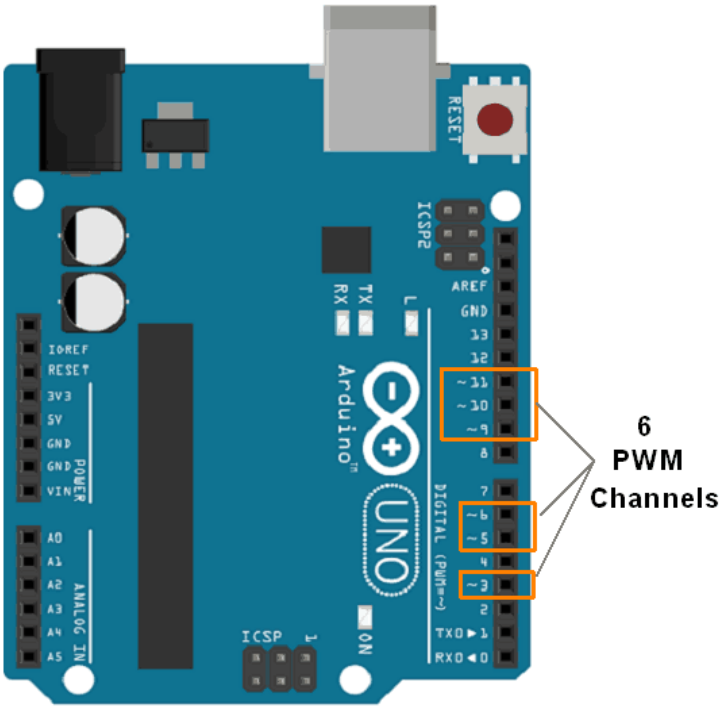
F=1/T

وكذلك الزمن فى PWM يساوى

T=1/F

اطراف PWM فى الاردوينو اونو :

وهى الاطرف الموجود عليها الرمز التالى ~



وهى الاطراف ارقام

3 – 5 – 6 – 9 – 10 – 11

والكود المستخدم فى الاردوينو للأعلان عن استخدام خاصية PWM هو

Analog write ( )

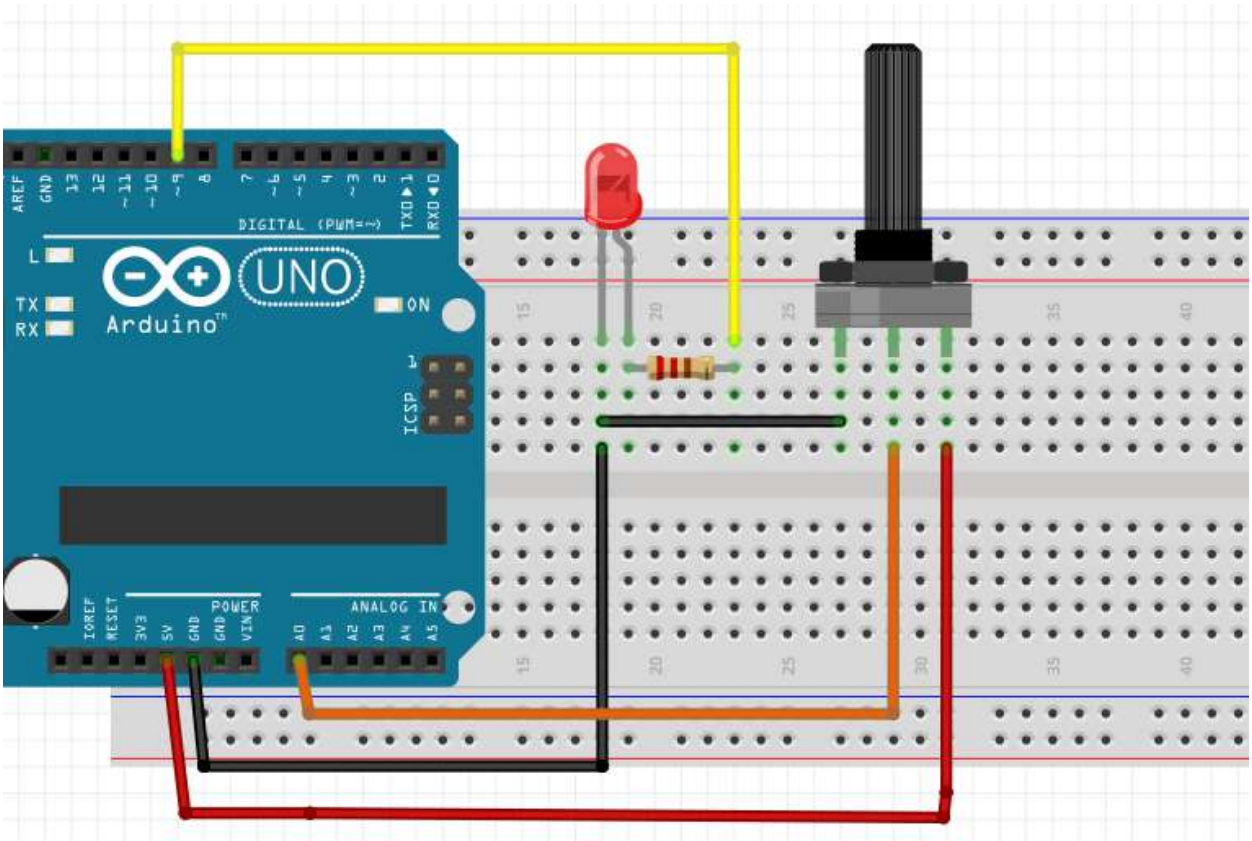
البرنامج (١)

التحكم فى شدة اضاءة ليد بواسطة مقاومة متغيرة

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – ليد – اردوينو اونو – مقاومة ٢٢٠ اوم – مقاومة متغيرة ١٠ كيلو اوم او ٢,٢ كيلو اوم

دائرة التوصيل :



كود البرنامج :

```
int ledPin = 9;
int analogPin = 0;
int val ;

void setup()

{
 pinMode(ledPin, OUTPUT);
}
void loop()
{
  val = analogRead(analogPin);
  analogWrite(ledPin, val/4 );
}
```

Done compiling.

ملاحظة :

قيمة المقاومة المستخدمة ٢,٢ كيلو اوم

ما الفرق بين analog read and analog write ؟

Analog read تأخذ القيم من ٠ الى ١٠٢٣

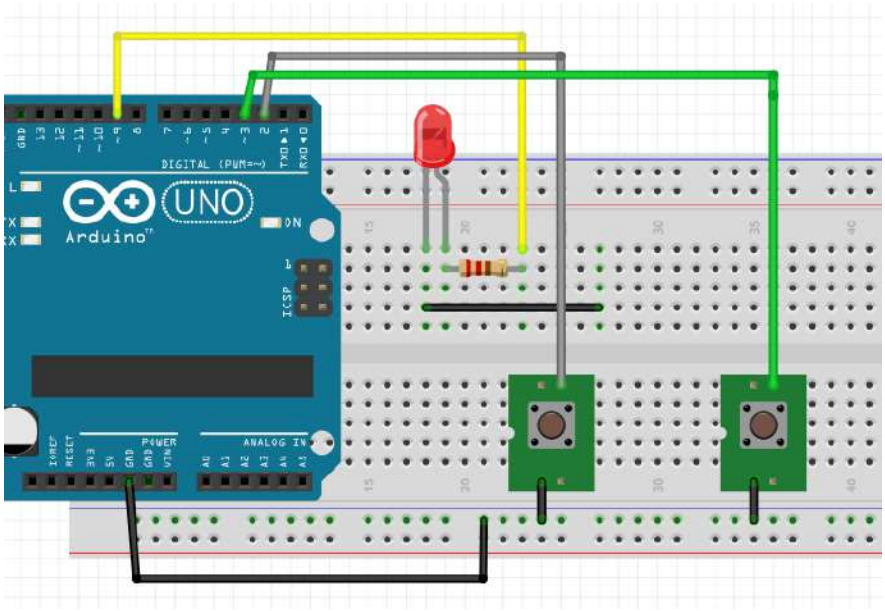
Analog write تأخذ القيم من ٠ الى ٢٥٥

لذلك تم قسمة val الموجودة بالكود على ٤

البرنامج (٢)

مفتاحين يتم التحكم من خلالهم فى شدة اضاءة الليد بحيث اذا تم الضغط على المفتاح الاول يزيد من شدة الاضاءة واذا تم الضغط على المفتاح الثانى يقلل من شدة الاضاءة .

دائرة التوصيل :

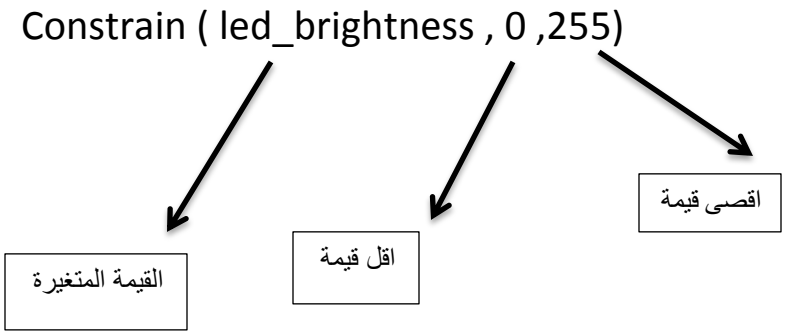


كود البرنامج :

```
int pin_sw1=2;
int pin_sw2=3;
int pinled=9;
int led_brightness;
void setup()
{
  pinMode(pin_sw1,INPUT);
  pinMode(pin_sw2,INPUT);
  pinMode(pinled,OUTPUT);
  digitalWrite(pin_sw1,HIGH);
   digitalWrite(pin_sw2,HIGH);
}
void loop()
{
  if(digitalRead(pin_sw1)==LOW)
  {
   led_brightness--;
  }
 else if(digitalRead(pin_sw2)==LOW)
 {
  led_brightness++;
 }
 led_brightness=constrain(led_brightness,0,255);
 analogWrite(pinled, led_brightness );
 delay(20);
}
```

Done compiling.

يجب ان تراعى اننا سوف نحصر قيمة PWM ما بين ٠ و ٢٥٥ لذلك سنستخدم الكود



اى ان حدود قيمة led_brightness ما بين ٠ و ٢٥٥

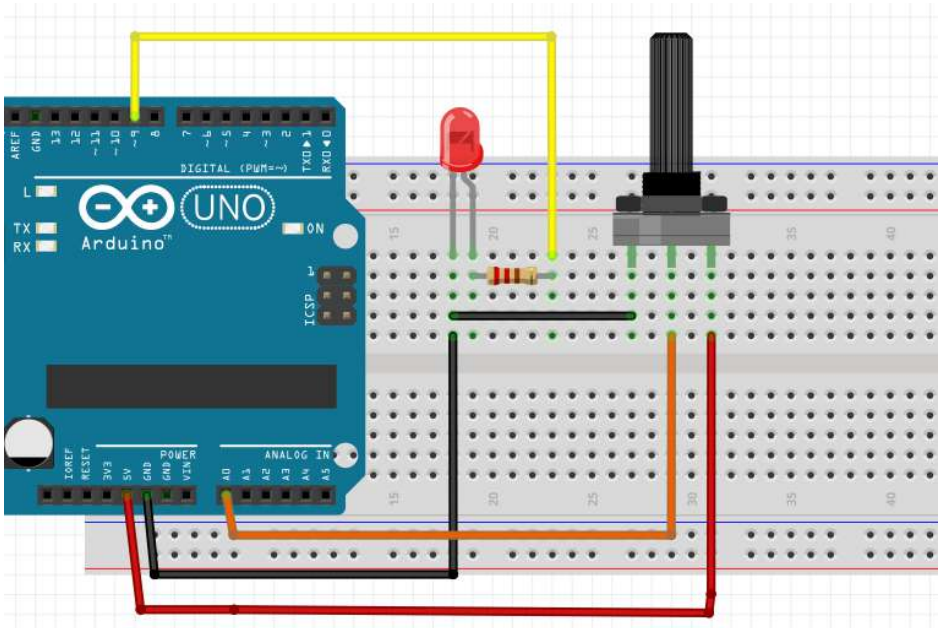
البرنامج (٣)

التحكم فى شدة اضاءة ليد بواسطة مقاومة متغيرة بأستخدام دالة map

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – اردوينو اونو – مقاومة متغيرة قيمتها ١٠ كيلو اوم – ليد – مقاومة ثابتة ٢٢٠ اوم

دائرة التوصيل :



كود البرنامج :

```
int ledPin = 9;
int analogPin = 0;
int led_brightness ;
int sensor_value;
void setup()

{
 pinMode(ledPin, OUTPUT);
}
void loop()
{
  sensor_value = analogRead(analogPin);
  led_brightness=map(sensor_value,0,1023,0,255);
  analogWrite(ledPin, led_brightness );
}
```

Done compiling.

دالة map ( )

تأخذ الشكل التالى

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

مثال لمتغير يتغير من القيمة ٠ الى القيمة ١٠٢٣ ونريد حصر العدد من ٠ الى ٢٥٥

Map (value , 0 , 1023 , 0 , 255)

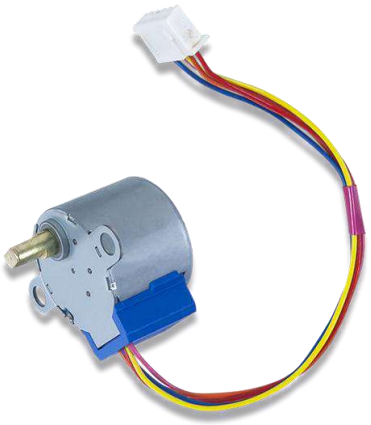
التحكم فى محركات التيار المستمر

يوجد العديد من محركات التيار المستمر اشهرهم :

Dc motor

Servo motor

Stepper motor



Brushless dc motor

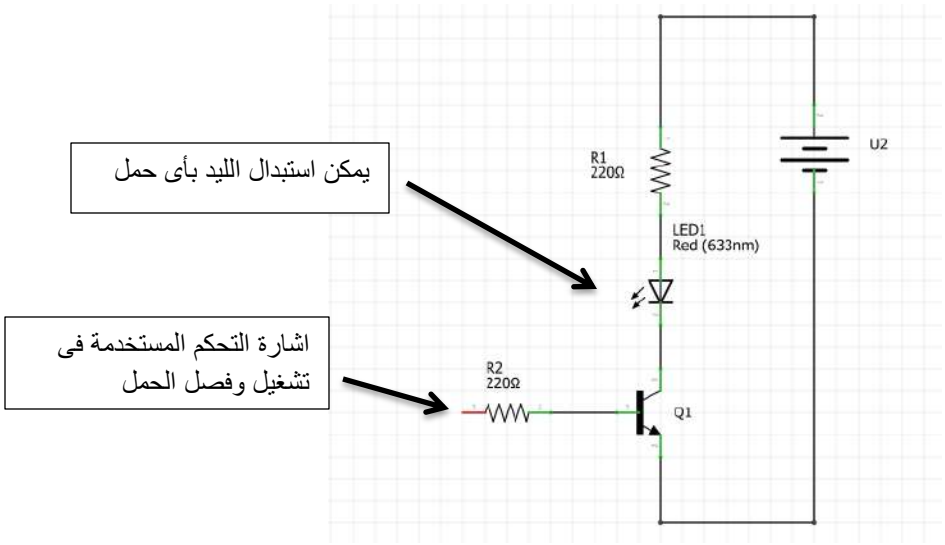
وفى هذا الكتاب سنستخدم نوعين فقط من المحركات وهما المحرك التيار المستمر العادى والاخر محرك السيرفو

طريقة تشغيل المحركات على لوحات الاردوينو :

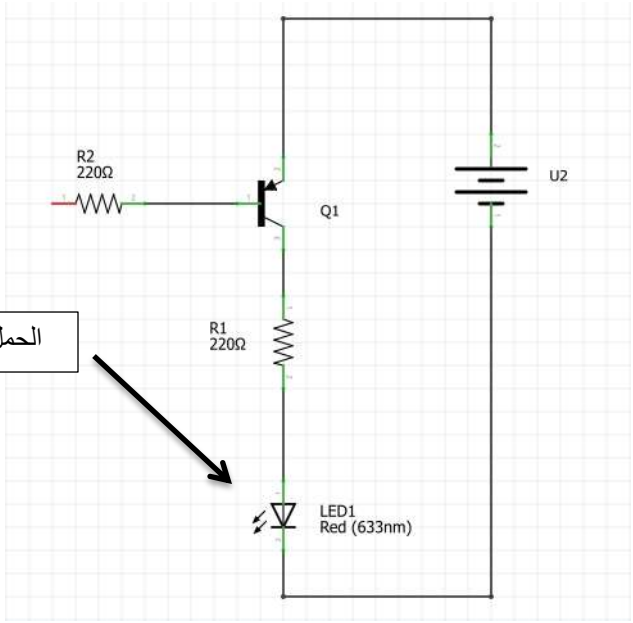
من المعلوم ان اى محرك يسحب تيار عالى يسمى تيار البدء من الممكن ان يؤدى هذا التيار الى تلف لوحة الاردوينو لذلك سنستخدم احد الطرق التالية للحفاظ على لوحة الاردوينو :

اولا : استخدام الترانزستور :

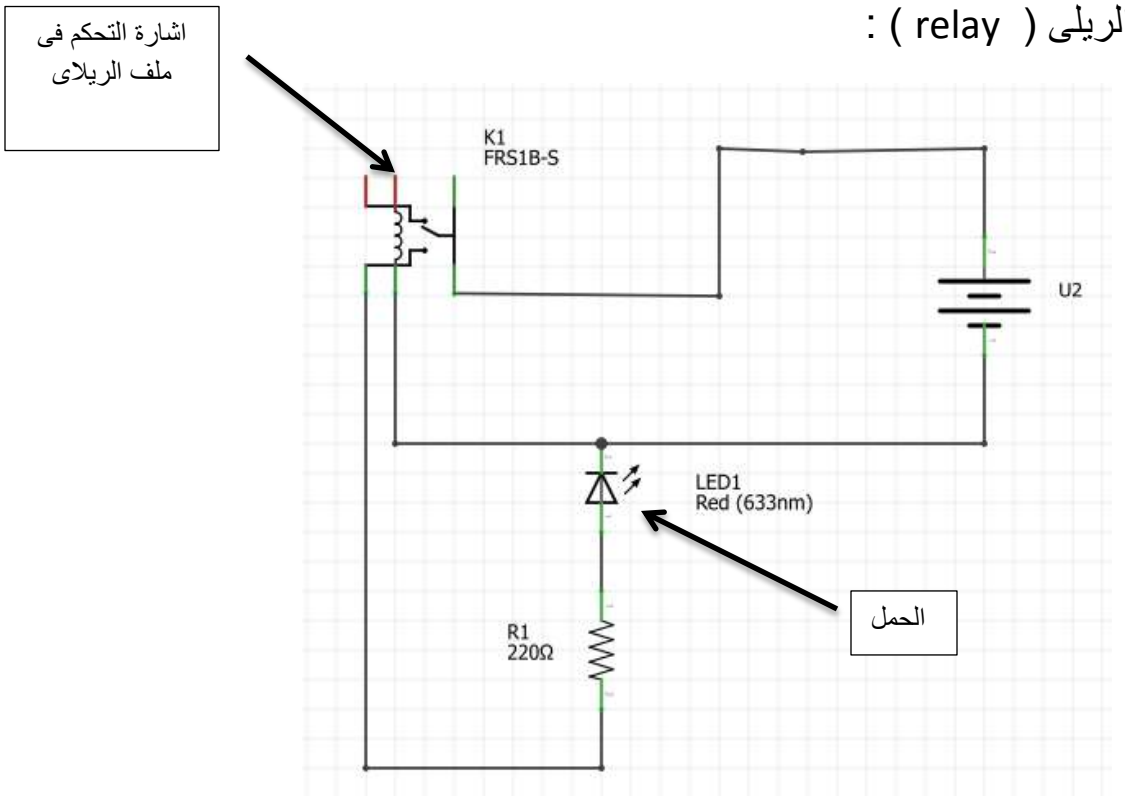
فى حالة استخدام ترانزستور من نوع NPN تكون دائرة التوصيل كالاتى :



فى حالة استخدام ترانزستور من نوع PNP تكون دائرة التوصيل كالاتى :



ثانيا : استخدام الريلى (relay) :



البرنامج (١)

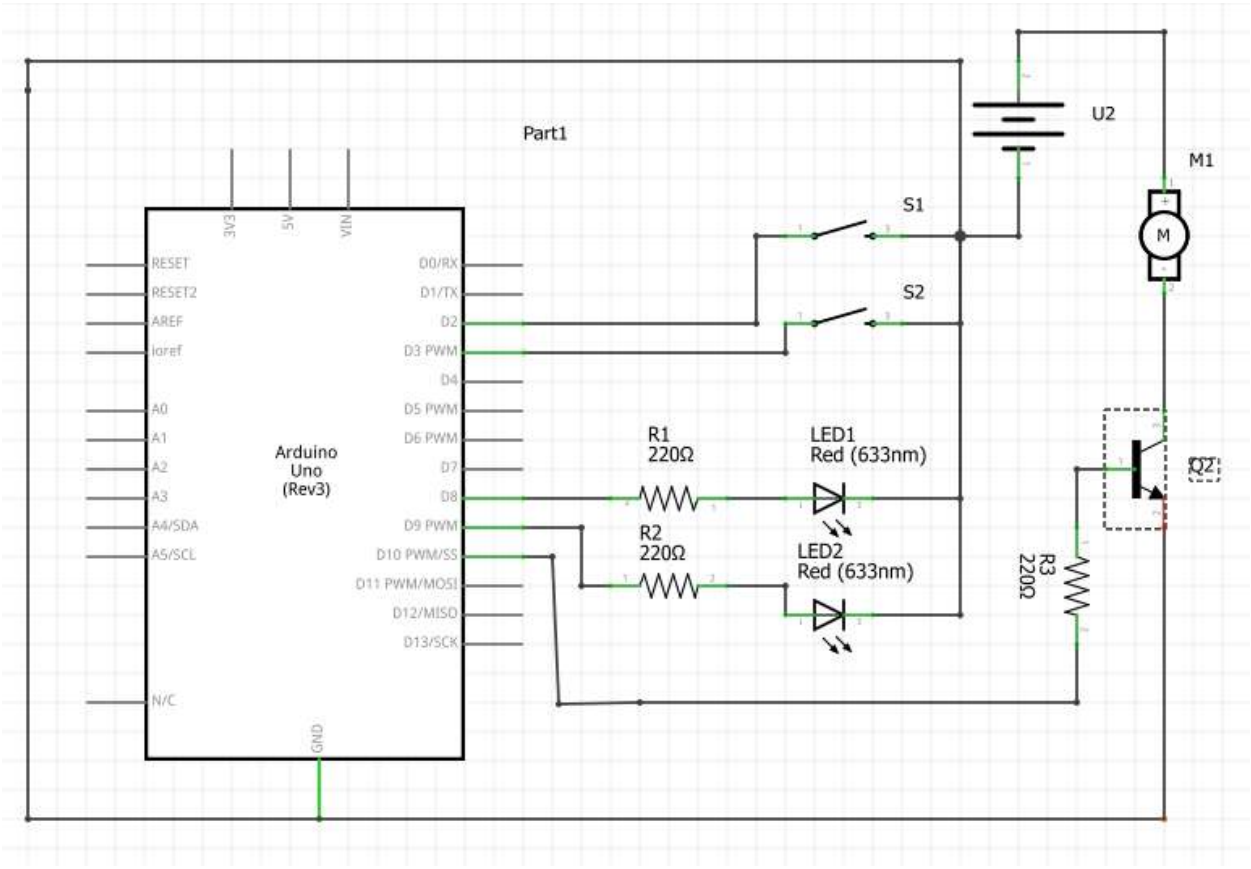
تشغيل واطفاء محرك بأستخدام مفتاحين احداهما للتشغيل والاخر للاطفاء مع ملاحظة وجود اثنين من الليد احداهما اخضر يضيء فى حالة تشغيل المحرك ويطفيء فى حالة اطفاء المحرك اما الليد الاحمر يضيء فى حالة اطفاء المحرك ويطفيء فى حالة تشغيل المحرك

المطلوب اعداد دائرة تقوم بالوظائف السابقة

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – ليد اخضر – ليد احمر – ثلاث مقاومات ٢٢٠ اوم – اردوينو اونو – محرك تيار مستمر – ترانزستور NPN – مفتاحين

دائرة التوصيل :



الترانزستور المستخدم 2N3904

كود البرنامج :

project_20 §

```
int pinled1= 8;
int pinled2 = 9;
int sw_pin1=2;
int sw_pin2=3;
int out_control_motor=10;
void setup(){
  pinMode(sw_pin1,INPUT);
  pinMode(sw_pin2,INPUT);
   pinMode(pinled1,OUTPUT);
   pinMode(pinled2,OUTPUT);
   pinMode(out_control_motor,OUTPUT);
    digitalWrite(sw_pin1,HIGH);
     digitalWrite(sw_pin2,HIGH);}
void loop(){
  if(digitalRead(sw_pin1)==LOW)
  {
    digitalWrite(pinled1,HIGH);
     digitalWrite(pinled2,LOW);
    digitalWrite(out_control_motor,HIGH);
  }
  else if (digitalRead(sw_pin2)==LOW)
  {
    digitalWrite(pinled1,LOW);
     digitalWrite(pinled2,HIGH);
    digitalWrite(out_control_motor,LOW);
  }
}
```

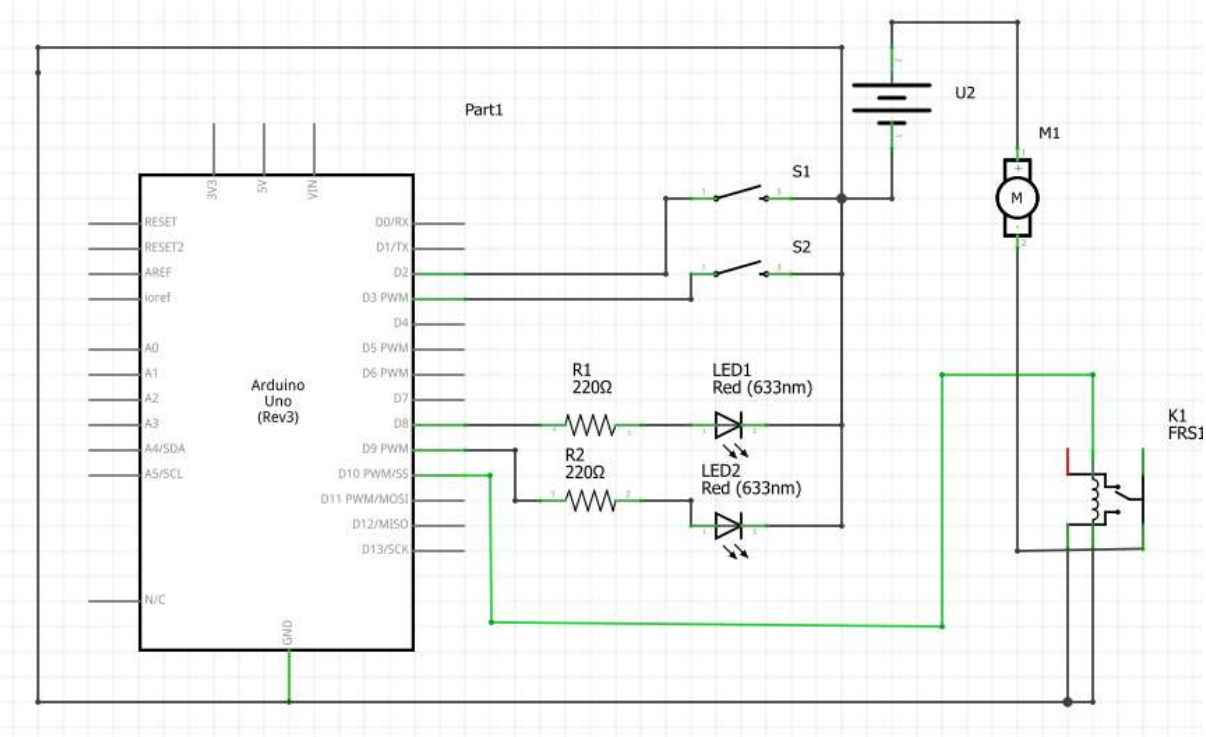
Done compiling.

ثانيا : تنفيذ نفس البرنامج السابق ولكن بأستخدام الريلاى :

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – ليد اخضر – ليد احمر – مقاومتين ٢٢٠ اوم – اردوينو اونو – محرك تيار مستمر – ريلاى – مفتاحين

دائرة التوصيل :

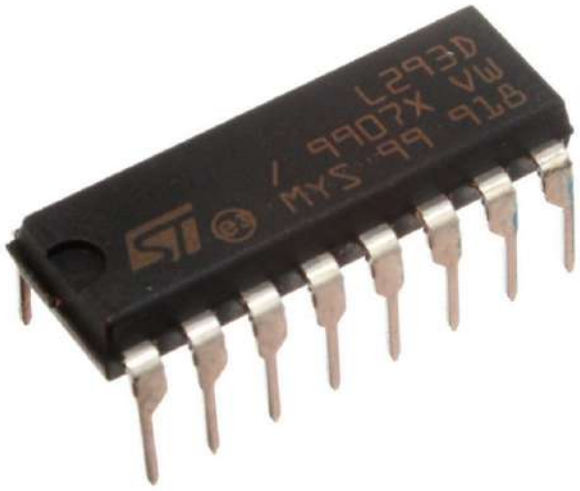


الفرق بين استخدام الترانزستور والريلاى ؟

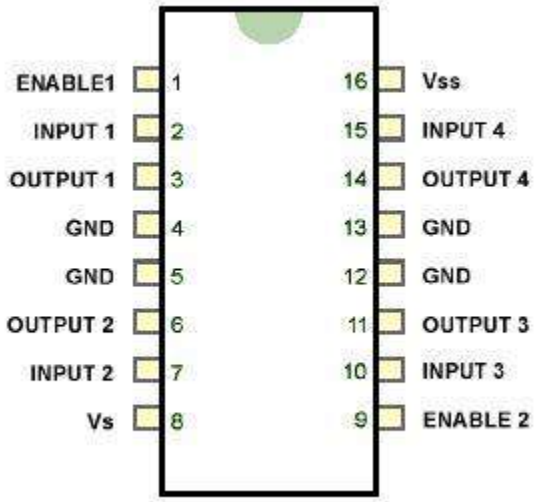
١. عزل الريلاى افضل من الترانزستور حيث يعمل على عزل دائرة التحكم عن المحرك
٢. يمكن استخدام الريلاى فى الدائر عالية القدرة الكهربية
٣. الترانزستور فى حالة الفصل والتوصيل اسرع من الريلاى

كيفية عكس اتجاة دوران محرك تيار مستمر :

وفى هذه الحالة يستخدم L293D

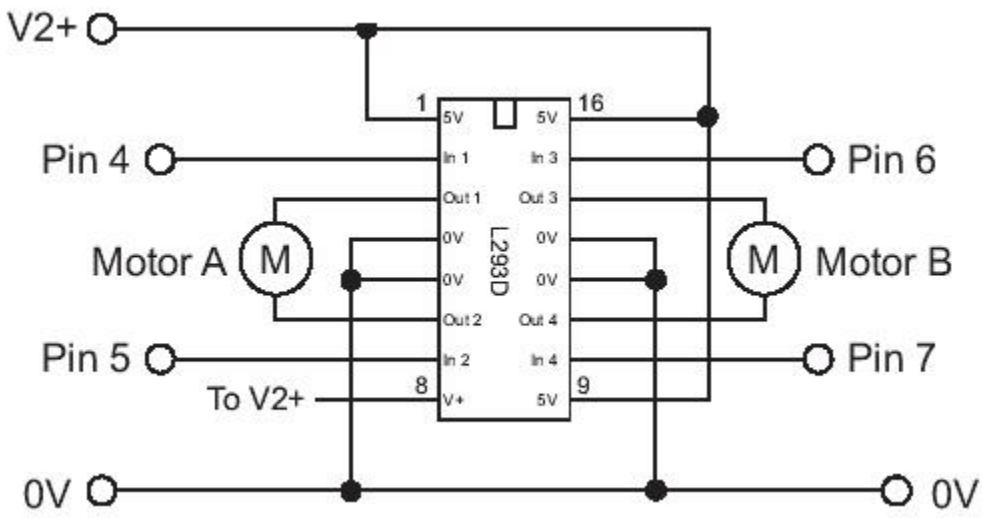


وصف الاطراف :



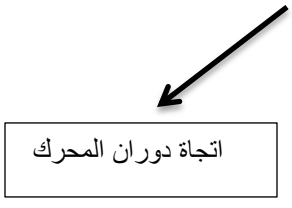
Pin out	Description
Enable 1	تفعيل تشغيل المحرك الاول
Input 1,2	اشارات التحكم فى المحرك الاول
Output 1, 2	توصيل المحرك الاول
vs	توصيل على الجهد(+)
vss	توصيل على الجهد(+)
Input 3 , 4	اشارات التحكم فى المحرك الثانى
gnd	توصيل الجهد(-)
Enable 2	تفعيل تشغيل المحرك الثانى
Output 3,4	توصيل المحرك الثانى

طريقة التوصيل :



اشارات التحكم :

Input 1	Input 2	Enable 1	Status
LOW	HIGH	HIGH	FORWARD
HIGH	LOW	HIGH	REVERSE



البرنامج (٢)

التحكم فى اتجاة دوران محرك حيث يتم التحكم بأستخدام ثلاث مفاتيح

الاول دوران المحرك فى اتجاة عقارب الساعة

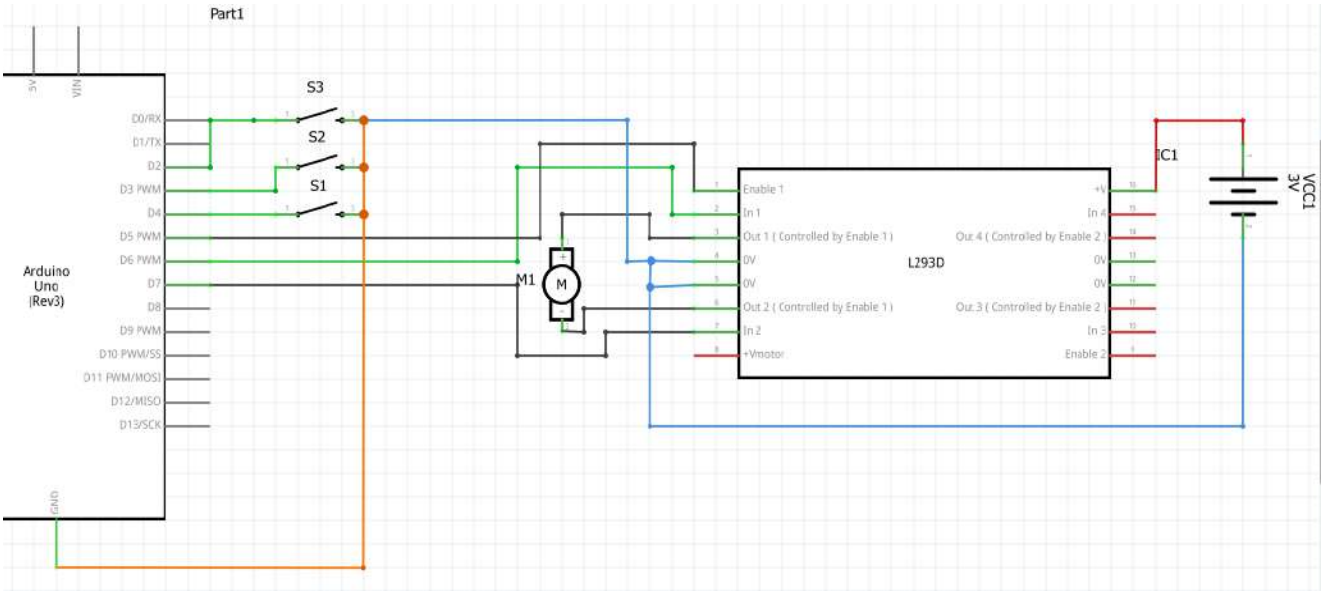
الثانى دوران المحرك فى اتجاه عكس عقارب الساعة

الثالث ايقاف المحرك

مكونات الدائرة :

لوحة اختبار – اسلاك توصيل - L293D – محرك – ثلاث مفاتيح

دائرة التوصيل :



كود البرنامج :

```
int sw_forword=2;
int sw_reverse=3;
int sw_stop=4;
int enable=5;
int input1=6;
int input2=7;
void setup(){
    pinMode(sw_forword,INPUT);
  pinMode(sw_reverse,INPUT);
  pinMode(sw_stop,INPUT);
   pinMode(enable,OUTPUT);
   pinMode(input1,OUTPUT);
   pinMode(input2,OUTPUT);
    digitalWrite(sw_forword,HIGH);
     digitalWrite(sw_reverse,HIGH);
     digitalWrite(sw_stop,HIGH);
}
void loop(){
   if(digitalRead(sw_forword)==LOW)
 {
    digitalWrite(enable,HIGH);
    digitalWrite(input1,HIGH);
     digitalWrite(input2,LOW);
  }
  else if (digitalRead(sw_reverse)==LOW)
  {
     digitalWrite(enable,HIGH);

    digitalWrite(input1,LOW);
     digitalWrite(input2,HIGH);
  }
  else if (digitalRead(sw_stop)==LOW)
  {
    digitalWrite(enable,LOW);
  }
}
```

السيرفو موتور :

وهو نوع من انواع محركات التيار المستمر الا انه من الممكن التحكم فى زوايا الدوران له

ويتميز هذا النوع من المحركات بالمميزات التالية :

١. يسحب تيار اقل
٢. عزم قوى
٣. دقيق فى الحركة
٤. يتوافر بأحجام مختلفة
٥. استمرارية فى العمل دون ارتفاع فى درجة حرارته

ويستعمل بكثرة فى الروبوتات والاجهزة التى تتطلب محركات ذات دقة عالية فى الحركة

ويتم التحكم فى زوايا دوران السيرفو موتور عن طريق تعديل النبضة المستخدمة على طرف التحكم من المحرك .

وينقسم السيرفو موتور الى نوعين من حيث زوايا الدوران :

فهناك نوع يمكنه التحرك من زواية ٠ الى ١٨٠ درجة اما النوع الاخر فيمكنه التحرك من زواية ٠ الى ٣٦٠ درجة

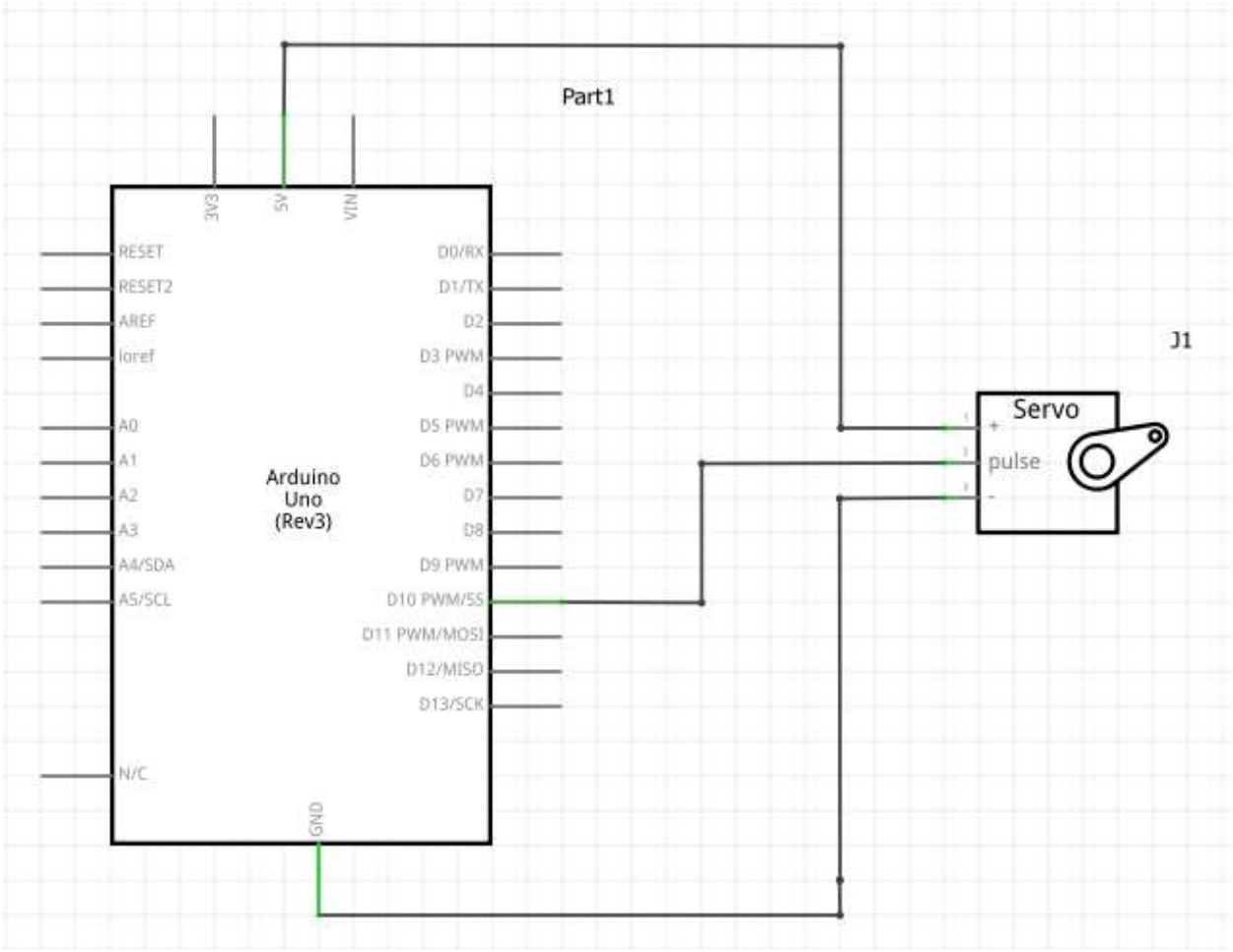
البرنامج (١)

التحكم فى حركة سيرفو موتور مابين زوايا ٩٠ , ١٨٠ , ١٣٥, ٩٠ , ٠ , ٤٥ درجة

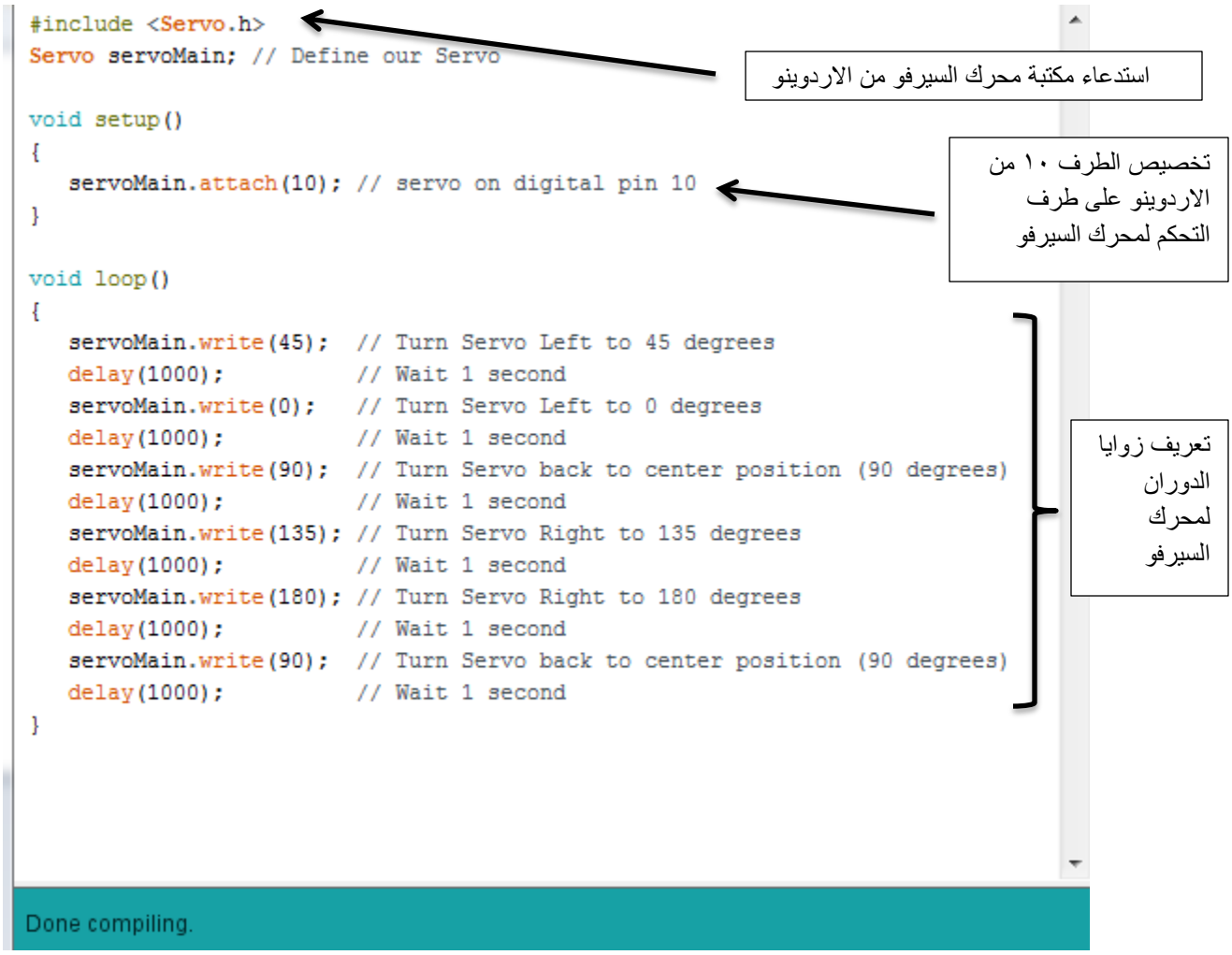
مكونات الدائرة :

لوحة اختبار – اسلاك توصيل – محرك سيرفو – اردوينو اونو

دائرة التوصيل :



كود البرنامج :



البرنامج (٢)

التحكم فى زوايا دوران محرك سيرفو موتور عن طريق المراقب التسلسلى serial monitor

فعند كتابة ٠ يكون موضع زواية الدوران للمحرك هى ٠ درجة

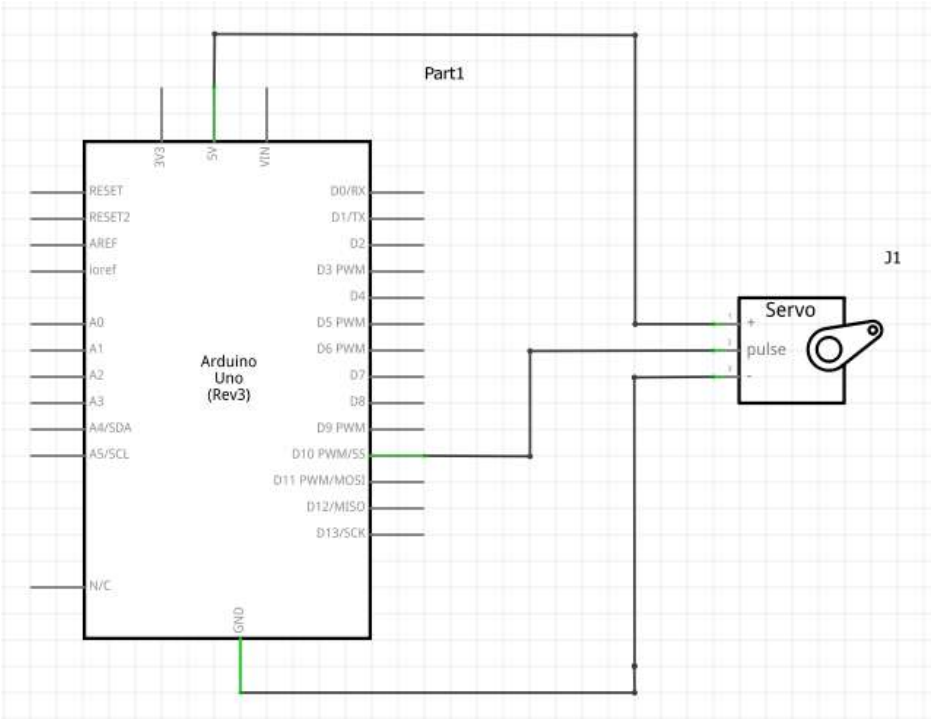
اما عند كتابة ١ يكون موضع زواية الدوران للمحرك هى ٤٥ درجة

اما عند كتابة ٢ يكون موضع زواية الدوران للمحرك هى ٩٠ درجة

اما عند كتابة ٣ يكون موضع زواية الدوران للمحرك هى ١٣٥ درجة

اما عند كتابة ٤ يكون موضع زواية الدوران للمحرك هى ١٨٠ درجة

دائرة التوصيل :



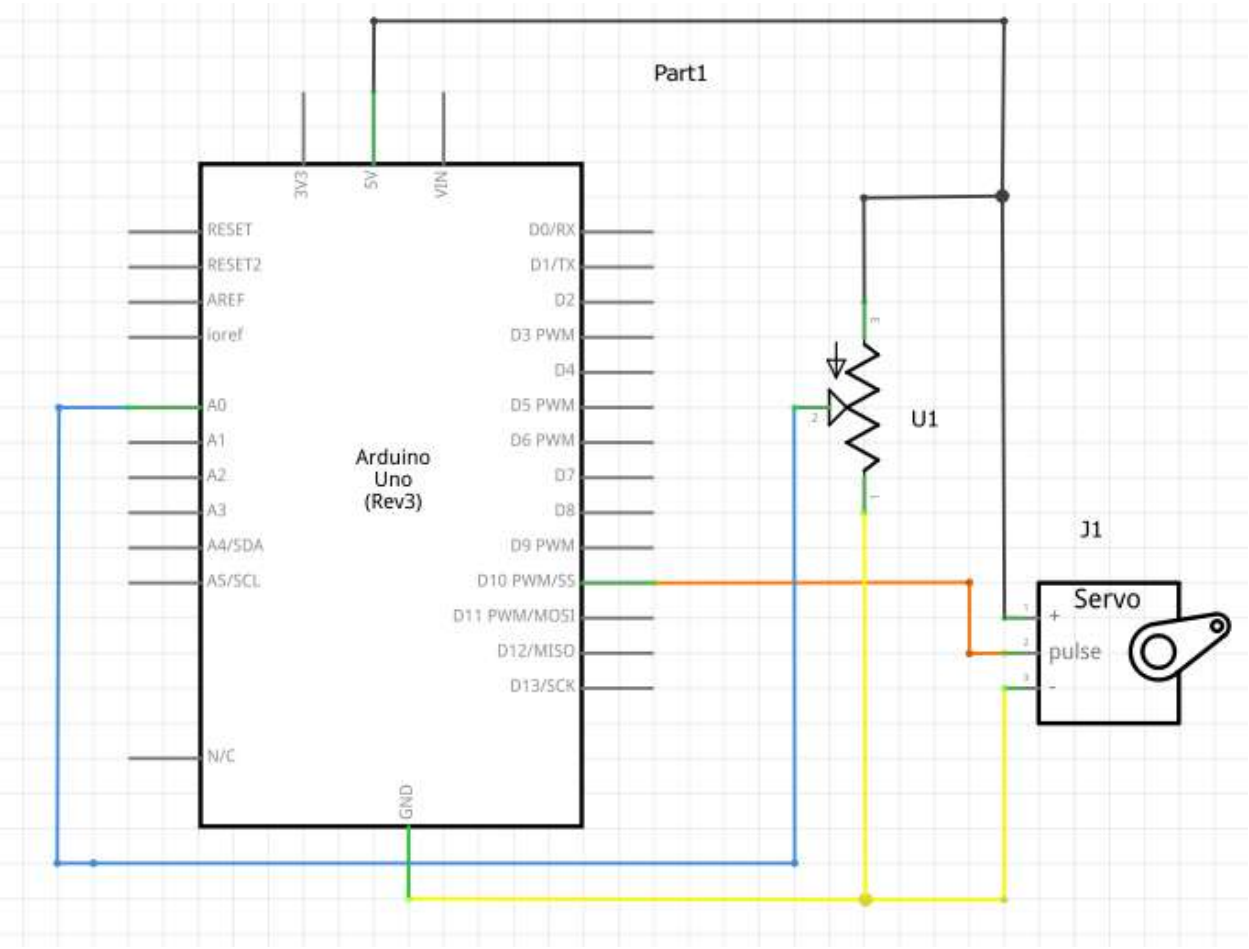
كود البرنامج :

```
#include <Servo.h>
Servo servoMain; // Define our Servo
int value;
void setup()
{
   servoMain.attach(10); // servo on digital pin 10
   Serial.begin(9600);
}
void loop()
{
  value=Serial.read();
  if(value=='0')
  {
   servoMain.write(0);   // Turn Servo Left to 0 degrees
            // Wait 1 second
    Serial.println("motor degree 0 ");
  }
  else if (value=='1')
  {
   servoMain.write(45);   // Turn Servo Left to 0 degrees
            // Wait 1 second
    Serial.println("motor degree 45 ");
  }
  else if (value=='2')
  {
   servoMain.write(90);   // Turn Servo Left to 0 degrees
          // Wait 1 second
    Serial.println("motor degree 90 ");
  }
   else if (value=='3')
  {
   servoMain.write(135);   // Turn Servo Left to 0 degrees
           // Wait 1 second
    Serial.println("motor degree 135 ");
  }
   else if (value=='4')
  {
   servoMain.write(180);   // Turn Servo Left to 0 degrees
          // Wait 1 second
    Serial.println("motor degree 180 ");
  }
}
```

البرنامج (٣)

التحكم فى دوران محرك سيرفو بواسطة مقاومة متغيرة

دائرة التوصيل :



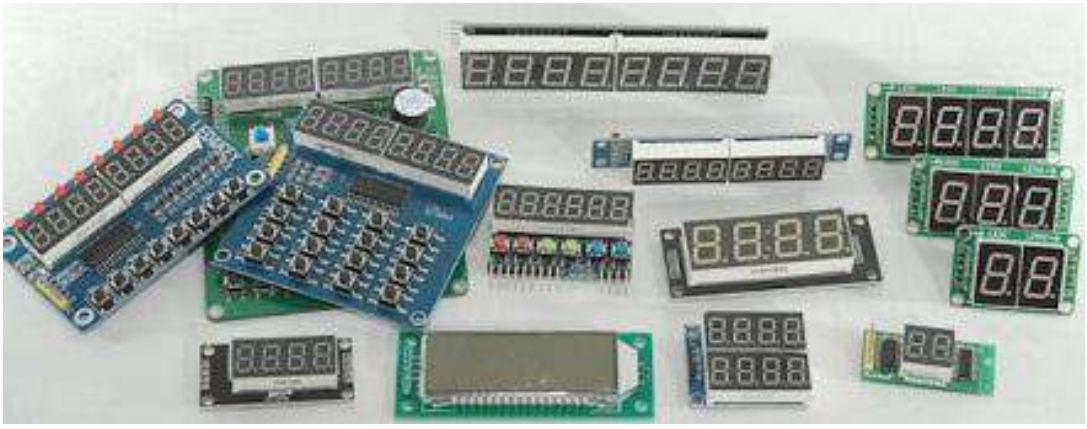
المقاومة المتغيرة المستخدمة قيمتها ١٠ كيلو اوم

كود البرنامج :

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup()
{
  myservo.attach(10);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180);     // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
}
```

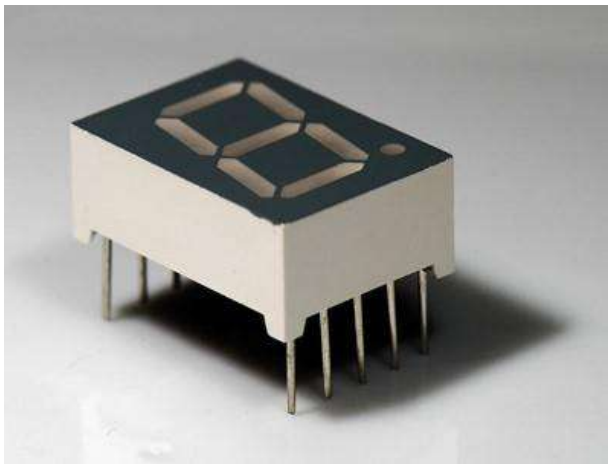
Done compiling.

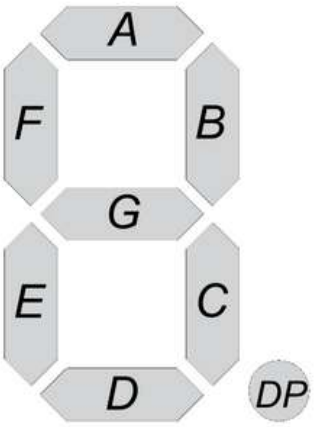
التعامل مع واجهات العرض



العارضة السباعية seven segment display

وهى عبارة عن عنصر الكترونى يحتوى على ٨ ليدات ويتم الحصول على الارقام من ٠ الى ٩ وذلك من خلال التحكم فى اضاءة واطفاء الليد





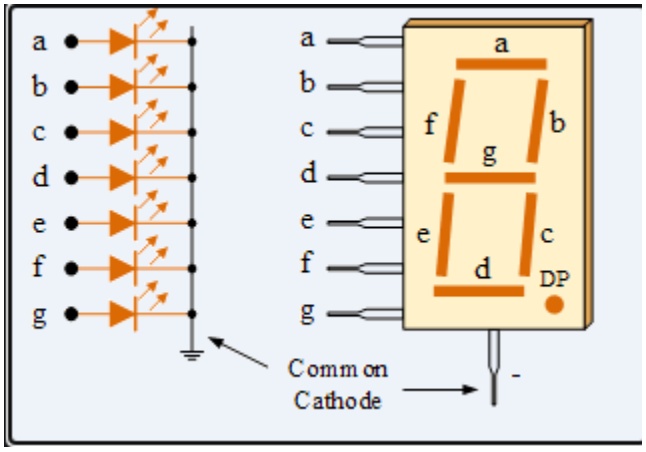
ويوجد نوعين من العارضة السباعية من حيث التوصيل :

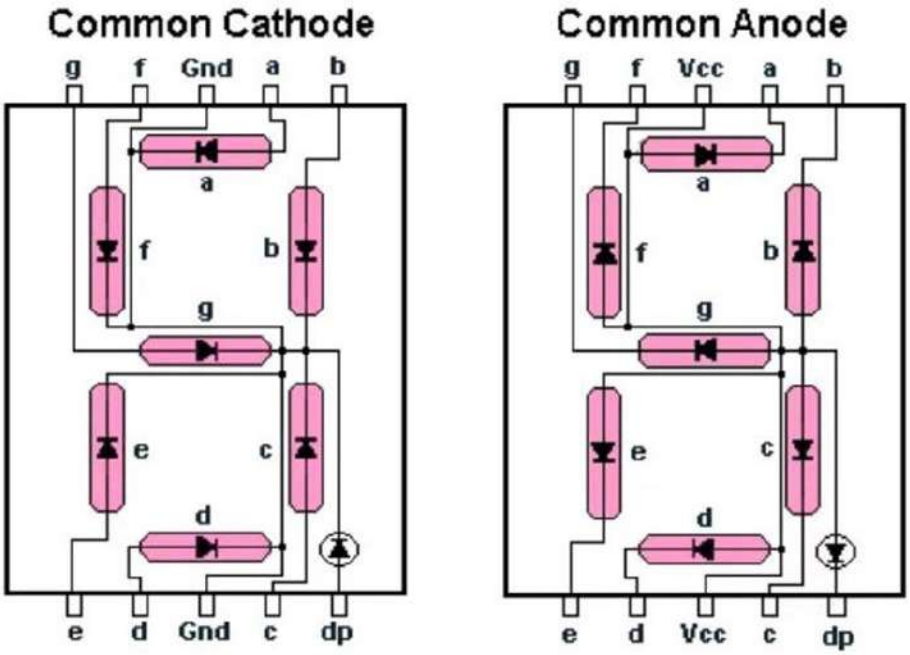
حيث النوع الاول يسمى common cathode

ويعمل كل ليد فى حالة اعطاء اشارة موجبة

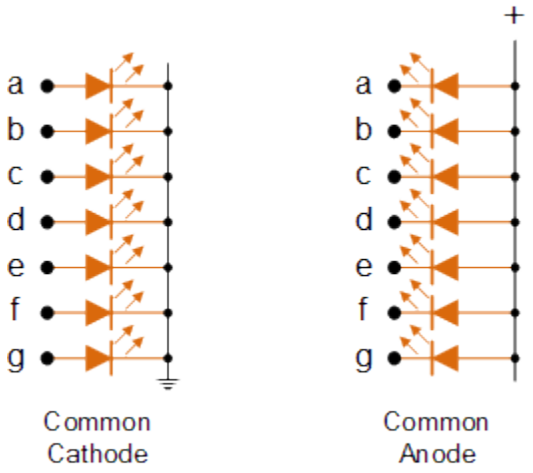
اما النوع الثانى يسمى common anode

ويعمل كل ليد فى حالة اعطاء اشارة سالبة





SSD Configuration

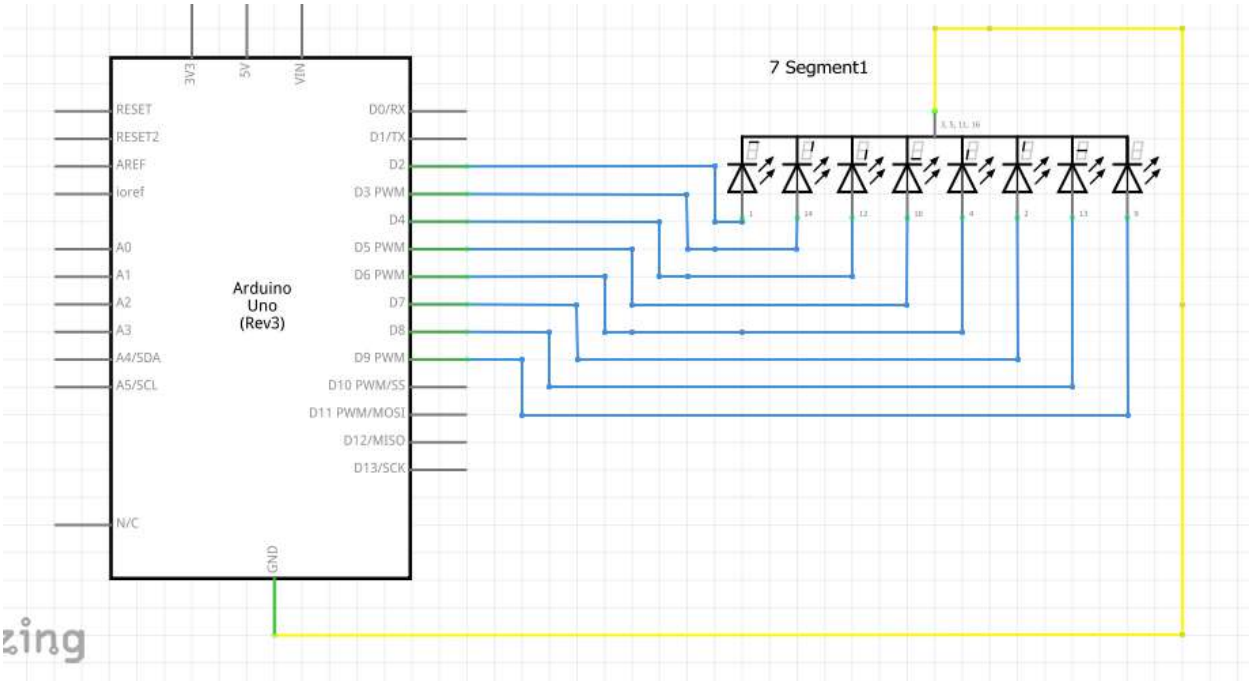


الجدول التالى يوضح طريقة تشغيل الليدات لتمثيل الارقام على العارضة السباعية

a	b	c	d	e	f	g	number
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9

البرنامج (١)

انشاء عداد تنازلى من ٩ الى ٠ باستخدام العارضة السباعية



كود البرنامج :

```
byte seven_seg_digits[10][7] = { { 1,1,1,1,1,1,0 },  // = 0
                                 { 0,1,1,0,0,0,0 },  // = 1
                                 { 1,1,0,1,1,0,1 },  // = 2
                                 { 1,1,1,1,0,0,1 },  // = 3
                                 { 0,1,1,0,0,1,1 },  // = 4
                                 { 1,0,1,1,0,1,1 },  // = 5
                                 { 1,0,1,1,1,1,1 },  // = 6
                                 { 1,1,1,0,0,0,0 },  // = 7
                                 { 1,1,1,1,1,1,1 },  // = 8
                                 { 1,1,1,0,0,1,1 }   // = 9
                                                   };
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  writeDot(0);  // start with the "dot" off
}
void writeDot(byte dot)
{
  digitalWrite(9, dot);
}
void sevenSegWrite(byte digit)
{
  byte pin = 2;
  for (byte segCount = 0; segCount < 7; ++segCount) {
    digitalWrite(pin, seven_seg_digits[digit][segCount]);
    ++pin;
  }
}
void loop()
{
  for (byte count = 10; count > 0; --count)
  {
   delay(1000);
   sevenSegWrite(count - 1);
  }
  delay(4000);
}
```

مصفوفة تحتوى على احتمالات تشغيل العارضة السباعية لتمثيل الارقام

البرنامج (٢)

انشاء عداد تصاعدى من ٠ الى ٩ بالعارضة السباعية

دائرة التوصيل :

نفس دائرة التوصيل السابقة

كود البرنامج:

```
byte seven_seg_digits[10][7] = { { 1,1,1,1,1,1,0 },  // = 0
                                 { 0,1,1,0,0,0,0 },  // = 1
                                 { 1,1,0,1,1,0,1 },  // = 2
                                 { 1,1,1,1,0,0,1 },  // = 3
                                 { 0,1,1,0,0,1,1 },  // = 4
                                 { 1,0,1,1,0,1,1 },  // = 5
                                 { 1,0,1,1,1,1,1 },  // = 6
                                 { 1,1,1,0,0,0,0 },  // = 7
                                 { 1,1,1,1,1,1,1 },  // = 8
                                 { 1,1,1,0,0,1,1 }   // = 9
                                                   };
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  writeDot(0);  // start with the "dot" off
}

void writeDot(byte dot)
{
  digitalWrite(9, dot);
}
```

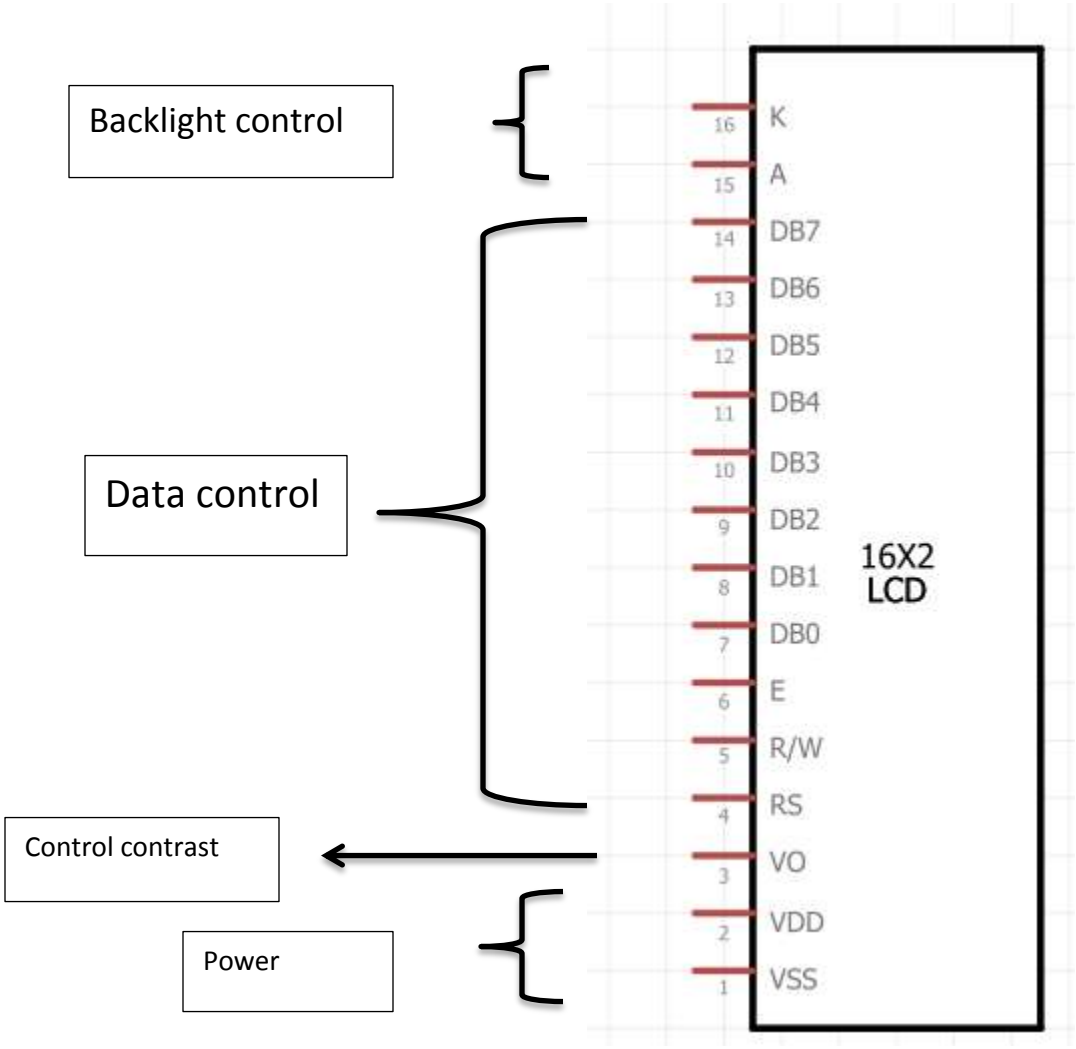
```
void sevenSegWrite(byte digit)
{
  byte pin = 2;
  for (byte segCount = 0; segCount < 7; ++segCount)
  {
    digitalWrite(pin, seven_seg_digits[digit][segCount]);
    ++pin;
  }
}

void loop()
{
  for (byte count = 0; count <10; ++count)
  {
   delay(1000);
   sevenSegWrite(count );
  }
  delay(4000);
}
```

التعامل مع الشاشات LCD

الشاشة المستخدمة فى الامثلة القادمة LMB162AFC

نقول ان الشاشة ١٦X ٢ اى ان عدد الاسطر المستخدمة على الشاشة سطرين وفى كل سطر يمكن استخدام ١٦ حرف

التعرف على اطراف التوصيل للشاشة



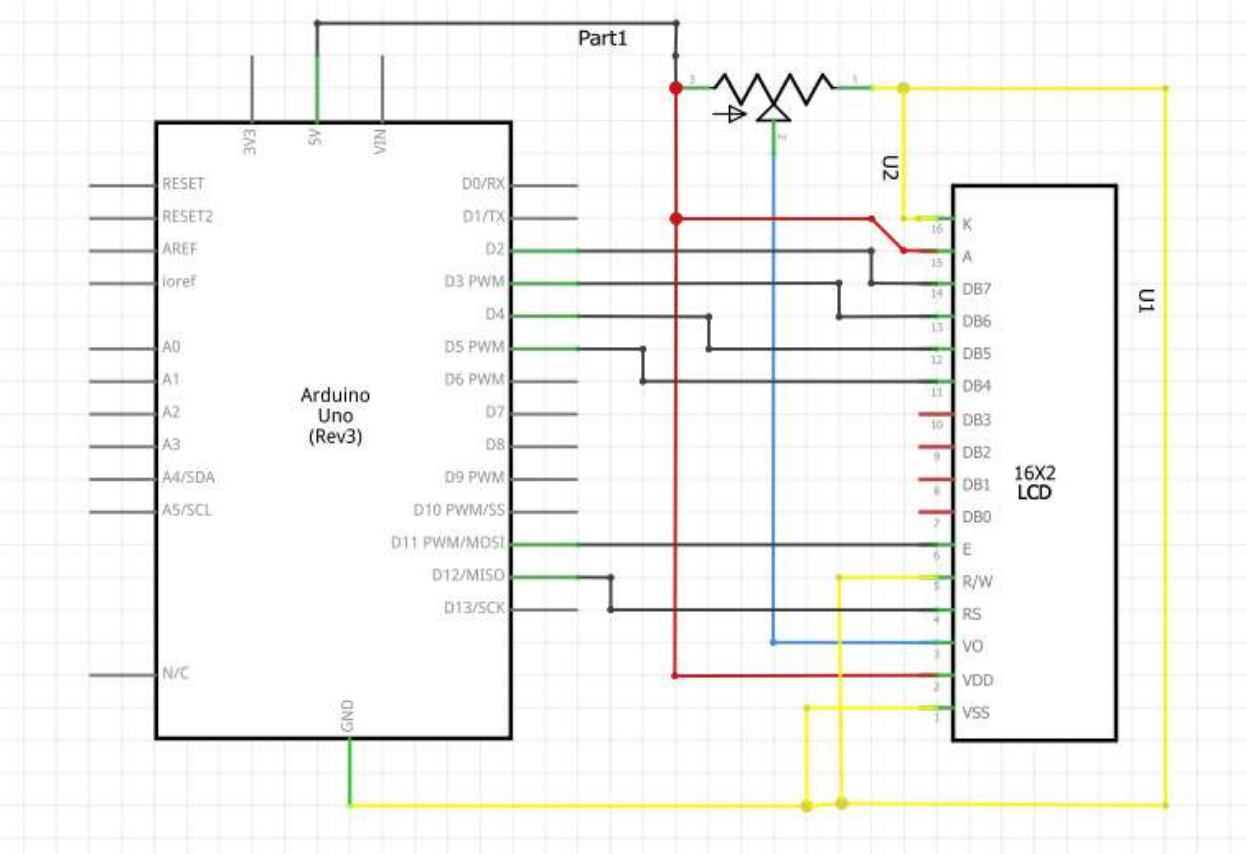
البرنامج (١)

طباعة جملة hello world على السطر الاول اما على السطر الثانى طباعة thank you

مكونات الدائرة :

لوحة اختبار – اردوينو اونو – اسلاك توصيل – مقاومة متغيرة ١٠ كيلو اوم – شاشة LMB162AFC

دائرة التوصيل :



كود البرنامج :

```
// include the library code:
#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);
  lcd.print("thank you !");
}
```

استخدام مكتبة LCD

تعريف الاطراف المستخدمة مع الاردوينو والشاشة

Done compiling.

البرنامج (٢)

فى السطر الاول يتم طباعة الاعداد من ٠ الى ٩ ثم عمل ازاحة وطباعة للارقام مرة اخرى فى السطر الثانى من الشاشة

دائرة التوصيل :

نفس دائرة التوصيل السابقة

كود البرنامج :

```
// include the library code:
#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);
}
void loop() {
  // set the cursor to (0,0):
  lcd.setCursor(0, 0); // print from 0 to 9:
  for (int thisChar = 0; thisChar < 10; thisChar++) {
    lcd.print(thisChar);
    delay(500);
  }
  // set the cursor to (16,1):
  lcd.setCursor(16, 1); // set the display to automatically scroll:
  lcd.autoscroll();
  // print from 0 to 9:
  for (int thisChar = 0; thisChar < 10; thisChar++) {
    lcd.print(thisChar);
    delay(500);
  }
  // turn off automatic scrolling
  lcd.noAutoscroll();
  // clear screen for the next loop:
  lcd.clear();
}
```

Done compiling.

البرنامج (٣)

طباعة الاعداد من ٠ الى ١٠٠ على الشاشة

دائرة التوصيل :

نفس دائرة التوصيل السابقة

كود البرنامج :

```
// include the library code:
#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);
}
void loop() {
  // set the cursor to (0,0):
  lcd.setCursor(0, 0);
  // print from 0 to 9:
  for (int thisChar = 0; thisChar < 100; thisChar++)
  {
    lcd.print(thisChar);
    delay(500);
     lcd.clear();
  }
}
```

Done compiling.

البرنامج (٤)

استخدام serial monitor فى ارسال بيانات الى شاشة LCD

حيث السطر الاول يظهر فيه جملة pc connection

اما السطر الثانى يتم طباعة ما سيتم كتابتة فى شاشة المراقب التسلسلى

دائرة التوصيل :

نفس الدائرة السابقة

كود البرنامج :

```
#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("pc connection");
  Serial.begin(9600);
}
void loop() {
  if(Serial.available())
  {
    lcd.setCursor(0, 1);
    lcd.print("                ");
    delay(100);
    lcd.setCursor(0, 1);
    while(Serial.available() > 0)
    {
      lcd.write(Serial.read());
    }
  }
}
```

Done compiling.

البرنامج (٥)

يتم كتابة اى شيء على المراقب التسلسلى وارساله على الشاشة LCD بشرط عند وصول الحد الاقصى من الكلمات على الشاشة فيتم مسحها .

دائرة التوصيل :

نفس التوصيل السابق

كود البرنامج :

```
// include the library code:
#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);
  // initialize the serial communications:
  Serial.begin(9600);
}
void loop()
{
  // when characters arrive over the serial port...
  if (Serial.available())
  {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0)
    {
      // display each character to the LCD
      lcd.write(Serial.read());
    }
  }
}
```

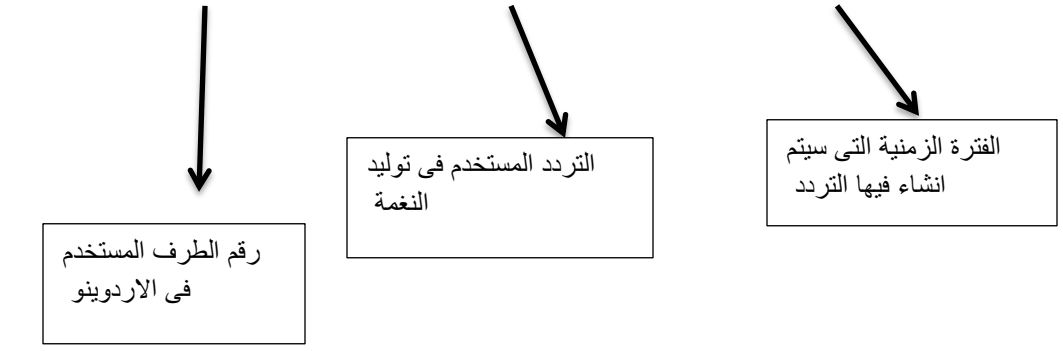
```
Done compiling.
```

التعامل مع الاصوات

يمكن انشاء النغمات بأستخدام الاردوينو بواسطة سماعة pizo

ولكى تولد النغمات بواسطة الاردوينو عليك بأستخدام كود tone والذى يأخذ الصورة البرمجية التالية :

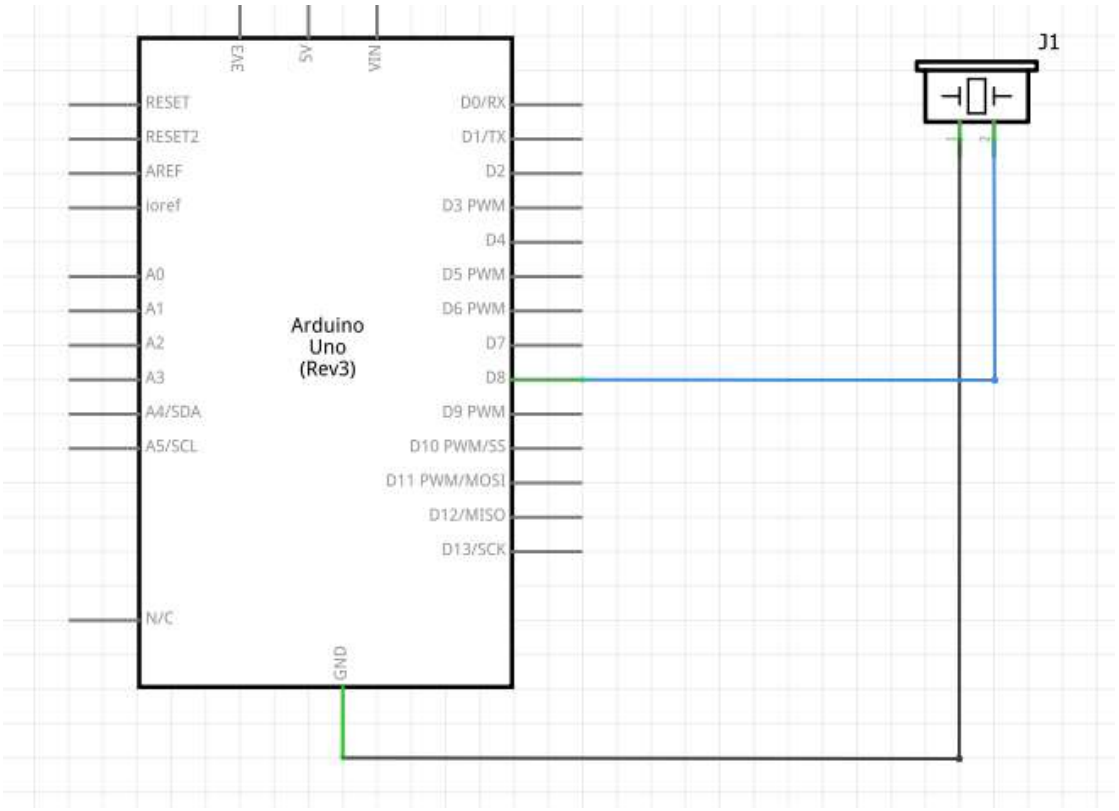
Tone (pin number , frequency in hertz , duration in m sec)



البرنامج (١)

انشاء نغمة صوتية

دائرة التوصيل :



كود البرنامج :

```
//Specify digital pin on the Arduino that the positive lead of piezo buzzer is attached.
int piezoPin = 8;

void setup()
{

}//close setup

void loop()
{
  tone(piezoPin, 1000, 500);
  //delay(1000);
}
```

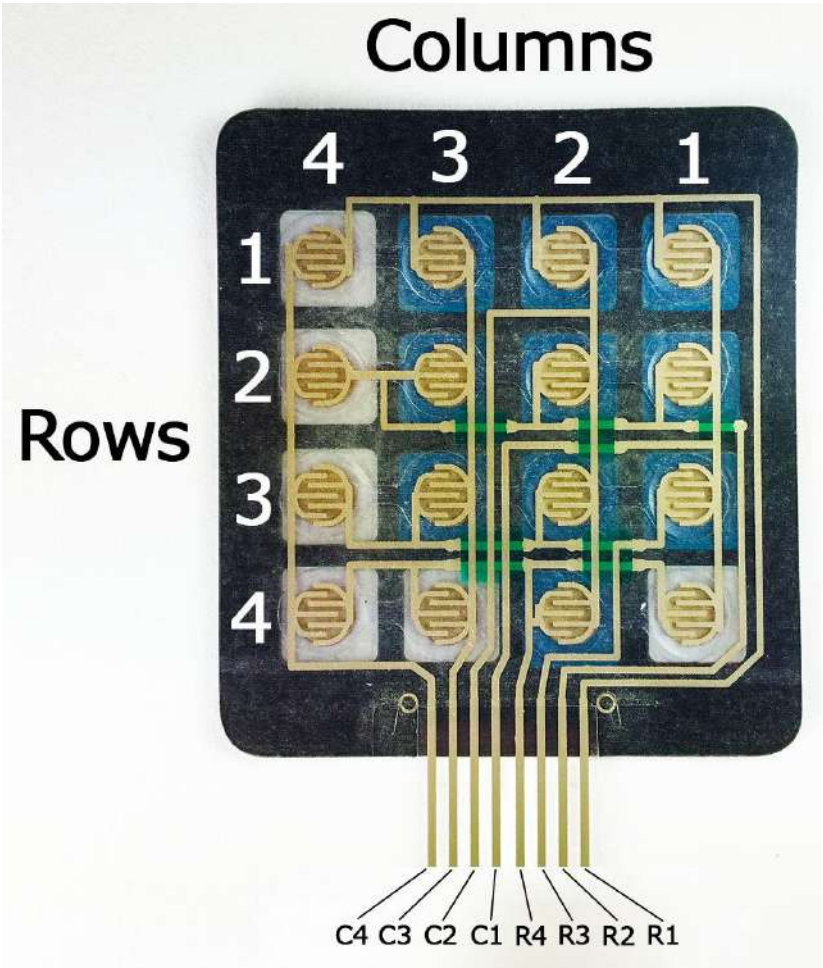
Done compiling.

يؤخذ فى الاعتبار الملاحظات التالية :

- عند توليد نغمة على الاردوينو يجب استخدام الاطراف التى تدعم PWM
- فى كل مرة يتم تغيير التردد تتغير النغمة
- كما يمكن انشاء نغمات متقطعة وذلك بتغير قيمة delay

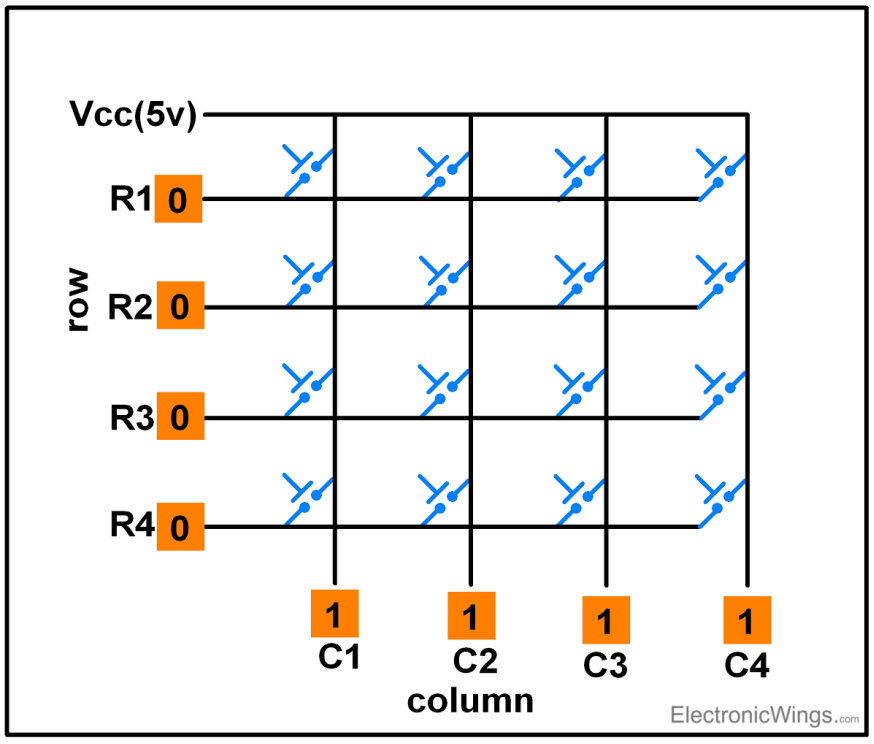
التعامل مع لوحة المفاتيح keypad

تتكون لوحة المفاتيح من ١٢ مفتاح مقسمة الى صفوف واعمدة كما بالشكل



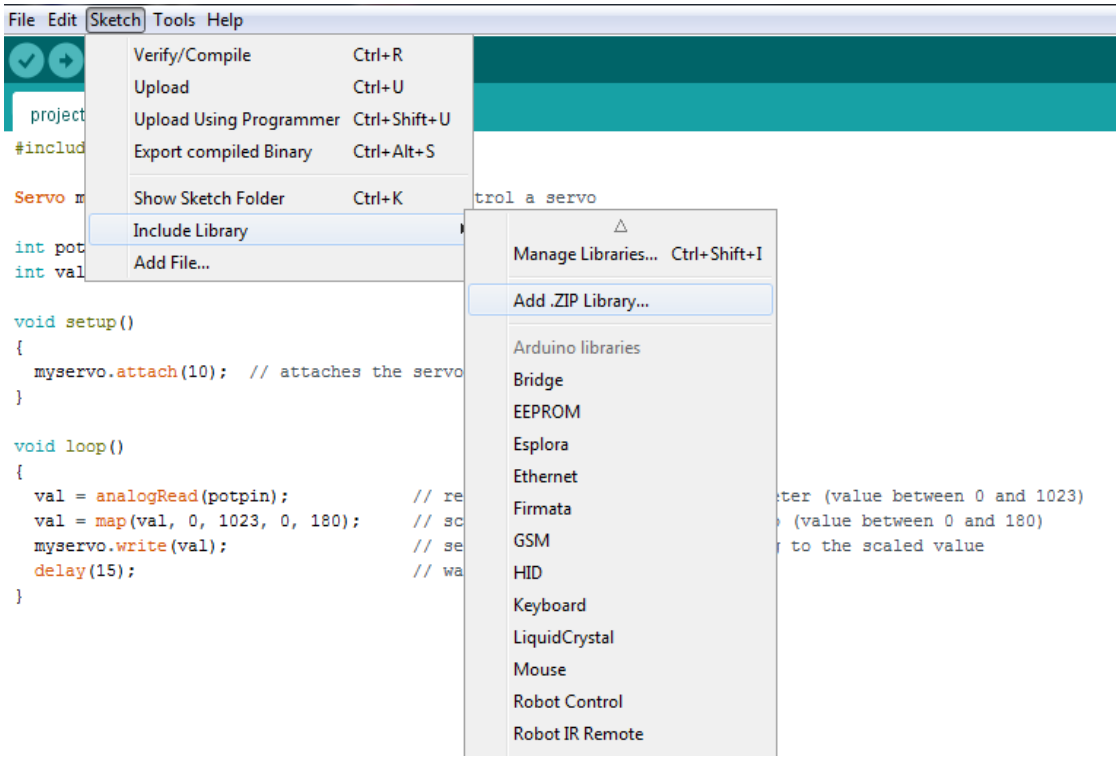
3x4 Keypad

4X4 Matrix Keypad



قبل البدء فى استخدام keypad عليك بتحميل مكتبة keypad.zip على الاردوينو

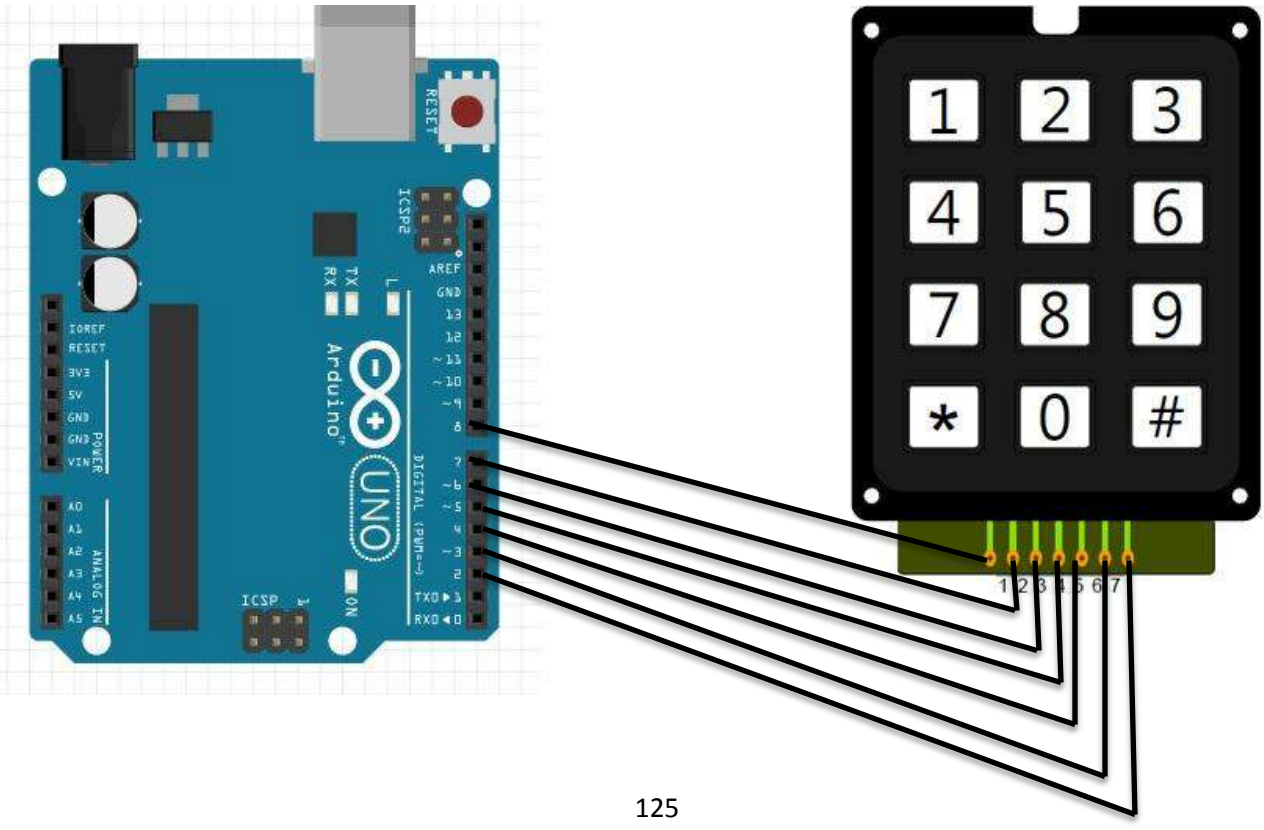
خطوات تحميل المكتبة كما بالصورة التالية :



البرنامج (١)

عند الضغط على اى رقم من ارقام keypad يتم طباعة رقم المفتاح فى serial monitor

دائرة التوصيل :



كود البرنامج :

```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'*','0','#'}
};
byte rowPins[ROWS] = { 5, 4, 3, 2 };
byte colPins[COLS] = { 8, 7, 6 };
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
void setup()
{
Serial.begin(9600);
}
void loop()
{
char key = keypad.getKey();

if (key!= No_KEY)
{
Serial.println(key);
}
}
```

مكتبة لوحة المفاتيح

البرنامج (٢)

انشاء برنامج يعمل على تشغيل محرك السيرفو بكلمة مرور فاذا كانت كلمة المرور صحيحة يتم اضاءة ليد اخضر وتشغيل محرك السيرفو اما اذا كانت كلمة المرور خاطئة يتم اضاءة الليد الاحمر ولا يتم تشغيل محرك السيرفو

دائرة التوصيل :

نفس الدائرة السابقة مع وضع الليد الاخضر على الطرف ١٣ للاردوينو بينما الليد الاحمر على الطرف ٩ اما اطراف السيرفو فيتم توصيل اشارة التحكم بالسيرفو على الطرف رقم ١١ للاردوينو

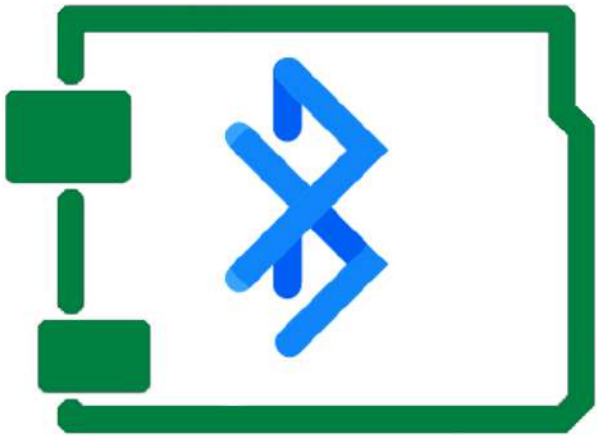
كود البرنامج :

```
#include <Keypad.h>
#include <Servo.h>
Servo servo_Motor;
char* password = "123";
int position = 0;
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'*','0','#'}
};
byte rowPins[ROWS] = { 5, 4, 3, 2 };
byte colPins[COLS] = { 8, 7, 6 };
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
int redPin = 9;
int greenPin = 13;
void setup()
{
pinMode(redPin, OUTPUT);
pinMode(greenPin, OUTPUT);
servo_Motor.attach(11);
setLocked(true);
}
void loop()
{
char key = keypad.getKey();
```

```
if (key == '*' || key == '#')
{
position = 0;
setLocked(true);
}
if (key == password[position])
{
position ++;
}
if (position == 3)
{
setLocked(false);
}
delay(100);
}
void setLocked(int locked)
{
if (locked)
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, LOW);
servo_Motor.write(11);
}
else
{
digitalWrite(redPin, LOW);
digitalWrite(greenPin, HIGH);
servo Motor.write(90);
```

```
}
}
```

التحكم باستخدام الاردوينو بواسطة البلوتوث



البلوتوث

فى هذا المثال سنستخدم HC – 05

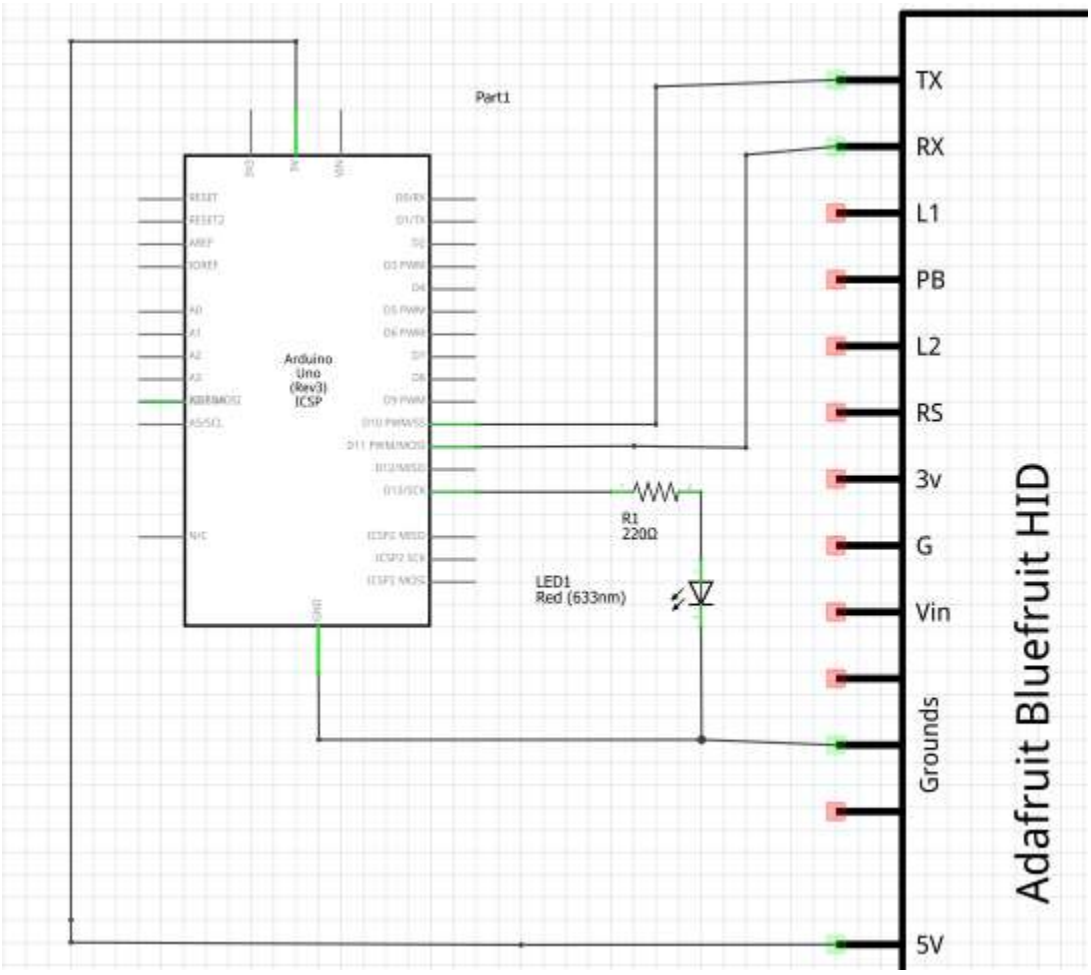
البرنامج (١)

التحكم فى اضاءة واطفاء ليد بواسطة الكمبيوتر او الموبايل وفى هذا المثال سيتم ربط بلوتوث الكمبيوتر مع وحدة البلوتوث hc05

ملاحظة عند استخدام الكمبيوتر فى التحكم فيمكنك انشاء برنامج بواسطة السى شارب او استعمال نافذة المراقب التسلسلى

اما عند استخدام الهاتف الجوال للتحكم فعندها ستختار البرامج الجاهزة من متجر جوجل

دائرة التوصيل :



كود البرنامج :

```
#include <SoftwareSerial.h>
SoftwareSerial BTserial(10, 11); // RX | TX
 int led=13;
 int BT_read;
void setup()
{
    Serial.begin(9600);
    Serial.println("Arduino is ready");
pinMode(led,OUTPUT);
    // HC-05 default serial speed for commincation mode is 9600
    BTserial.begin(9600);
}
void loop(){
 // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTserial.available())
    {
         BT_read = BTserial.read();
        Serial.write( BT_read);
    }
    if( BT_read=='0')
  {
    digitalWrite(led,LOW);
  }
  else if( BT_read=='1')
  {
     digitalWrite(led,HIGH);
  }
}
```

Done compiling.

بعد فتح نافذة المراقب التسلسلى يتم كتابة ١ ونلاحظ اضاءة الليد اما اذا تم كتابة ٠ سيتم اطفاء الليد