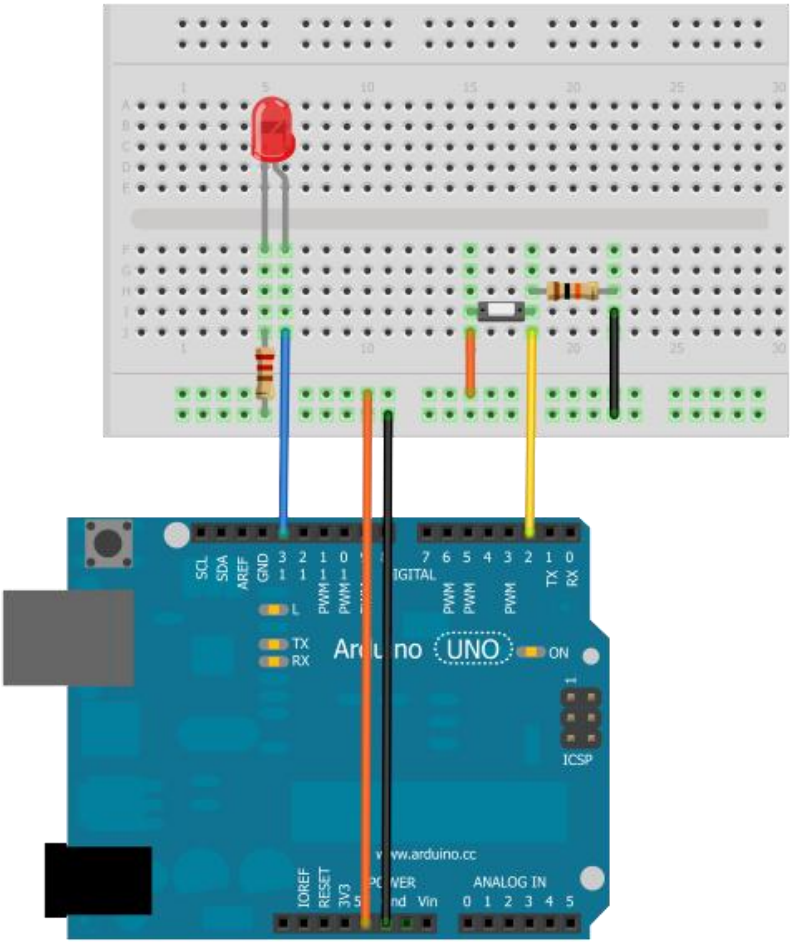
المثال الثانى: تطوير للمثال السابق ليعمل فقط عند ضغط مفتاح (سويتش) من نوع Push button

في هذا المثال سنقوم بتطوير الفكره السابقه و سنجعل الليد يعمل فى حاله اذا ضغط المستخدم على زر من نوع push button و هو اشهر انواع السويتشات المستخدمة في الأجهزة الإلكترونية.



مكونات المثال (2):

- بورده اردوينو Arduino uno
- Breadboard
- Push button
- Led 10 mm
- مقاومه 10 كيلو اوم
- مقاومه560 اوم
- بعض السلوك للتوصيل

قم بوضع السويتش على لوحه التجارب ثم وصل احدى طرفيه بالمدخل 5v باستخدام سلك على بورده اردوينو و الطرف الاخر بالمقاومه العشره كيلو اوم و من نفس النقطه قم بتوصيل سلك للمدخل رقم2 على **اردوينو** بورد و بعد ذلك قم بتوصيل طرف المقاومه الثانى بالمدخل الارضى .

ما الهدف من هذة الخطوة ؟؟

تعمل تلك الدائره على توفير دخل رقمي Digital input ، عند ضغط الزر تنطلق نبضه بقيمه 5 فولت و التى تعتبرها المتحكمه الدقيقه اشاره من نوع HIGH و عند ترك الزر ينفصل التيار الكهربى و يصبح الدخل يساوى صفر فولت و الذى تعتبره المتحكمه LOW و سوف نستغل هذا الامر لتعديل المثال الاول لكى يعمل فقط عندما يكون الدخل HIGH

الخطوة الثانية : كتابه الأكواد البرمجية التالية في Arduino IDE

نصيحه: قم بكتابه الأوامر البرمجيه بنفسك ولا تأخذها نسخ و لصق من الكتاب و ذلك حتي تدرب نفسك على كتابه الكود

```
// Example_2_Blinking_with_Switch

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

المرحله الاولى: تعريف المتغيرات و الثوابت

```
void setup ( )

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

 }
```

المرحله الثانيه: تعريف و تحديد المداخل و المخارج

```
void loop()

{
  val = digitalRead(buttonPin);

  if (val == HIGH)

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

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

المرحله الثالثه: كتابه اوامر البرنامج الرئيسيه

الخطوة الثالثة: قم بعمل تأكيد للكود البرمجى عن طريق الضغط على علامه تأكيد Verify و بعد ان تتأكد من ان البرنامج صحيح ولا يحتوى على اخطاء برمجيه و تم تحويله الى الصوره التنفيذيه بنجاح، قم بعد ذلك برفعه للمتحكمه الدقيقه عن طريق زر رفع Upload

شرح الكود البرمجى .....

المرحلة الاولى: قمنا بتعريف متغير اسمه **ledPin** و الذى سيرمز الى Pin 13 و كذلك قمنا بتعريف المتغير **buttonPin** و الذى سيرمز الى السويتش الموجود على المدخل الرقمى السابع و قمنا بتعريف المتغير **val** و الذى سوف نستخدمه لاحقا لتخزين حاله الدخل على المدخل الرقمى الثانى **pin 2**

```
int ledPin = 13;
int buttunPin = 2;
int val = 0;
```

المرحلة الثانية: فى هذه المرحلة جعلنا المتحكمة تعامل **pin 13** و التي يرمز لها باسم **ledPin** على أنها مخرج و سنستخدمه لتشغيل الليد الموصل به، ثم قمنا بجعل المتحكمة تعامل المخرج **pin 2** على انه مدخل رقمى وظيفته استقبال الاشارات الرقميه **HIGH or LOW**

```
void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(inPin, INPUT);

}
```

المرحله الثالثه:

```
val = digitalRead(buttunPin);
```

في هذا الأمر ستقوم المتحكمة بقياس قيمه الجهد على المدخل pin 2 و تسجل القيمة في المتغير val

و يتم ذلك باستخدام الأمر **digitalRead( )** و اذا كان المستخدم ضغط على السويتش فأن القيمه ستكون 5 فولت = HIGH و ان لم يضغط ستكون القيمه صفر فولت = LOW

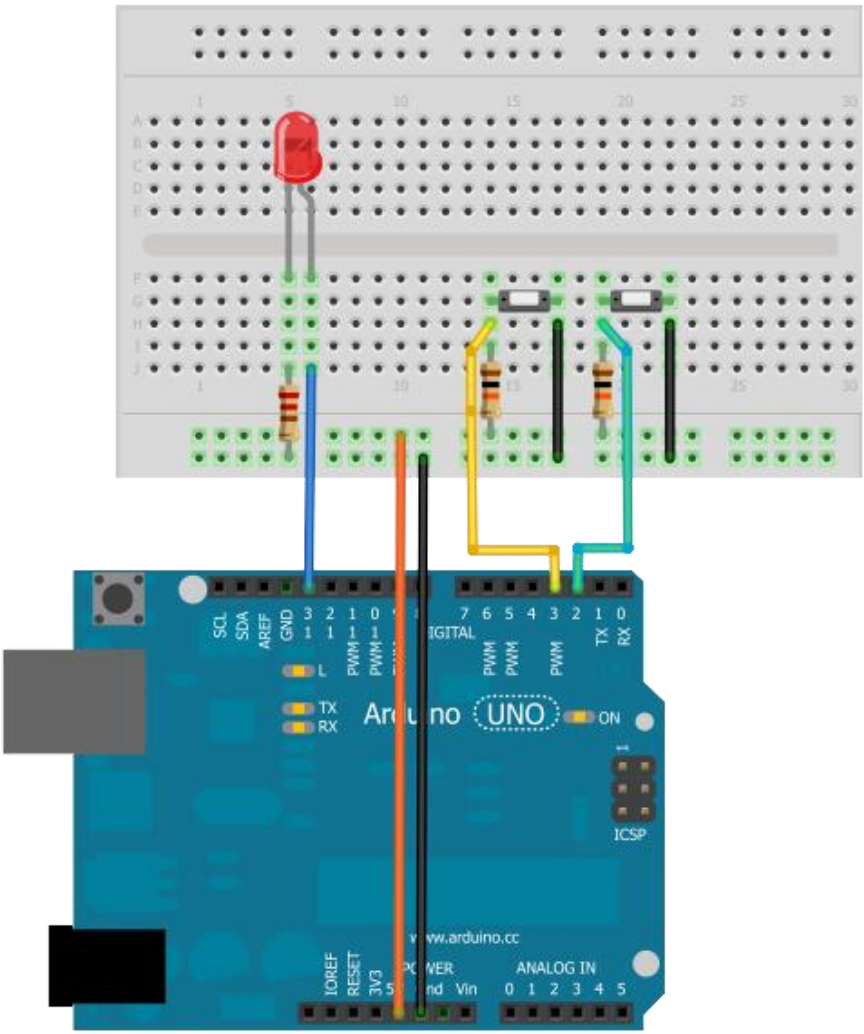
```
if (val == HIGH)
   {
digitalWrite(ledPin, HIGH);
delay(1000);
digitalWrite(ledPin, LOW);
delay(1000);
   }
else {digitalWrite(ledPin, LOW); }
```

فى هذا الجزء استخدمنا واحد من اشهر الاوامر فى لغات البرمجه و هو الامر

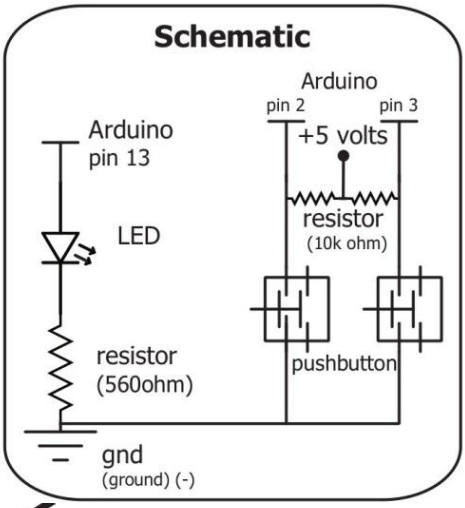
if….else (اذا = .. مادون ذلك)، يستخدم ذلك الامر لعمل مقارنات و جعل المتحكمه تفعل شئ ما اذا حدث شئ اخر محدد و فى هذا المثال استخدمنا الامر if كالتالى :

- ستقيس المتحكمه قيمه الجهد على المدخل السابع و تخزن القميه فى المتغير **val**
- اذا كانت القميه تساوى HIGH و التى تعنى 5 فولت ستقوم المتحكمه بتشغيل الليد لمده ثانيه و اطفائه لمده ثانيه و هكذا باستمرار.
- ما لم تكن القيمه تساوى HIGH و الجهد المطبق على المدخل يساوى صفر ستقوم المتحكمه بتطبيق جهد يساوى صفر على الليد و بذلك لن يعمل الليد و سيظل منطفئ.

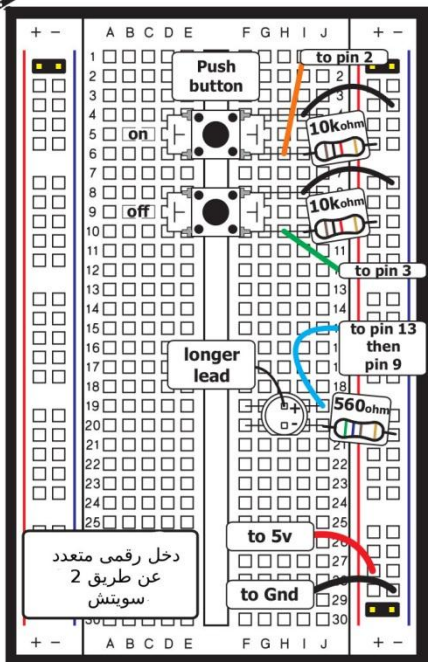
المثال الثالث: تطوير المثال السابق يعمل بـ2 سويتش من نوع Push Button واحد من اجـل التشـغيلON و الثانى من اجل الاغلاق OFF



مكونات المثال(3):

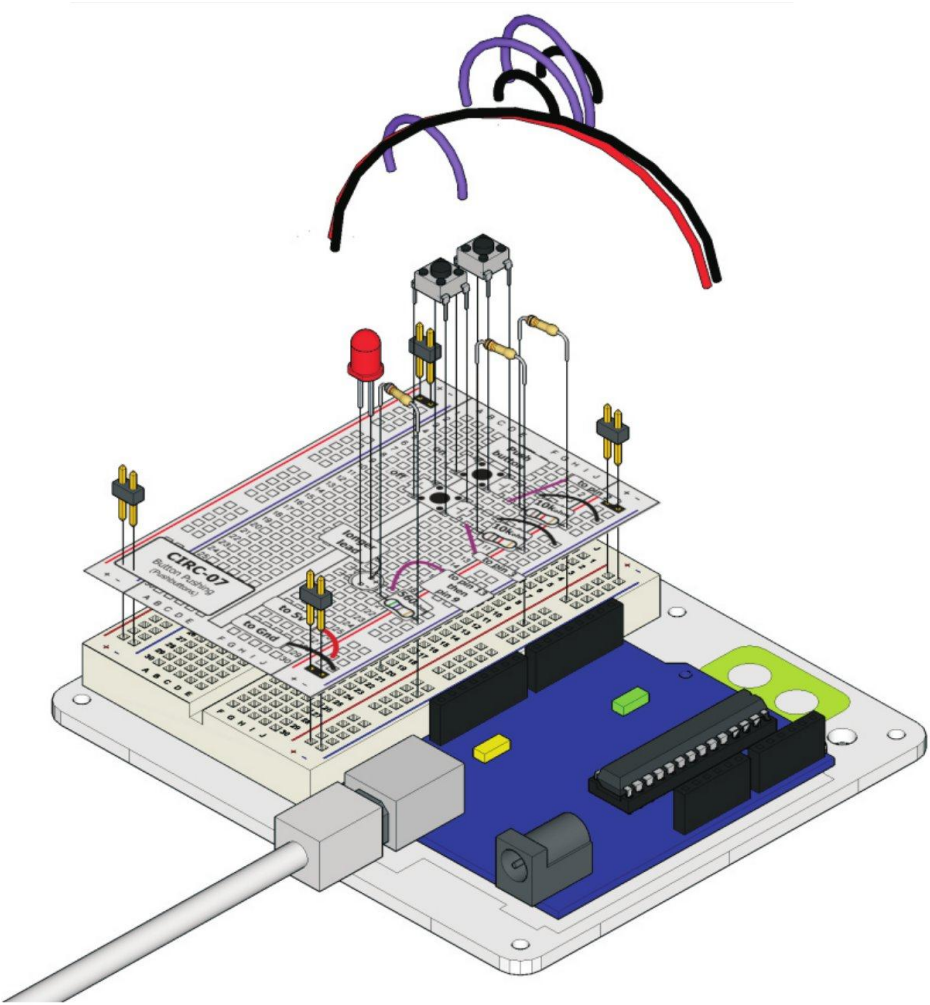


- Arduino uno
- Breadboard
- Led 5 mm
- عدد2 مقاومه قيمه 10 كيلو اوم
- مقاومه واحده قيمه 560 اوم
- عدد 2 سويتش
- اسلاك توصيل
- نسخه مطبوعه من الورقه المساعده



فى هذا المثال سندمج ما تعلمناه فى كلا المثالين الاول و الثانى ، سنستخدم 2 سويتش من نوع push button و سيتولى السويتش الاول عمليه تشغيل الليد و سيتولى السويتش الاخر عمليه اطفائه.

الخطوه الأولى : قم بطابعه ورقه المساعده و قصها ثم ثبتها على الـbreadboard باستخدام 4 من الـPin-Headers ثم قم بتوصيل المكونات الالكترونيه على الورقه المساعده كما فى الصوره



الخطوه الثانيه: من بيئه التطوير Arduino IDE قم بعمل ملف جديد من شريط الاوامر السريعه و اكتب الاوامر البرمجيه التاليه فى الجزء المخصص لكتابه الاكـواد و بعد الانتهاء ارفع البرنامج الى المتحكمه الدقيقه على البورده و جرب الضغط على السويتش الاول و الثانى و شاهد ماذا سيحدث

```
//Example_3_On_Off_with_2_Switchs_1
const int ledPin= 13;
const int inputPin1 = 2;
const int inputPin2 = 3;
void setup ( )
        {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin1, INPUT);
  pinMode(inputPin2, INPUT);
        }
void loop ( )
        {
  if (digitalRead(inputPin1) == HIGH)
    {digitalWrite(ledPin, LOW); }
else if (digitalRead(inputPin2) == HIGH)
    {digitalWrite(ledPin, HIGH); }
        }
```

تحديد المداخل و المخارج للدائره

اوامر البرنامج الرئيسيه

في هذا المثال استخدمنا تعديل بسيط على الأمر if و هو أضافه الأمر **else if** بدلاً من **else** فقط و هو أمر يمكننا من أضافه اكثر من احتمال واحد في جمله **if** .

مثال:

```
If (you like science) {you should read books}
else if (you like sport) {you should play football}
else if (you like art) { you should take art lessons}
else { you should search for a hobby to do in your summer}
```

كما ترى في هذا المثال وضعنا 3 احتمالات و و خيار أخير كتالى :

- **اذا** كنت تحب العلوم **عليك أن** تقرأ بعض الكتب
- **أما اذا** كنت تحب الرياضة **عليك أن** تلعب كره القدم
- **أما اذا** كنت تحب الفن **عليك أن** تتعلم دروس الرسم
- **و اذا لم** يكن لديك هواية **عليك أن** تبحث عن هواية لتشغل بها وقت الإجازة الصيفية

و في البرنامج السابق استخدمنا احتمال أن المستخدم سوف يضغط على السويتش الأول فتقوم المتحكمة الدقيقة بتشغيل الدايود الضوئي و الاحتمال الثاني هو أن يضغط المستخدم السويتش 2 فتقوم المتحكمة بإطفاء الدايود الضوئي ... و هكذا

و الأن ما رأيك أن نعدل البرنامج قليل (^_^)

```
//Example_3_On_Off_with_2_Switchs_2
const int ledPin= 13;
const int inputPin1 = 2;
const int inputPin2 = 3;

void setup ( )
{
pinMode(ledPin, OUTPUT);
pinMode(inputPin1, INPUT);
pinMode(inputPin2, INPUT);
}

void loop ( )
{

if (digitalRead(inputPin1) == HIGH)
   {
digitalWrite(ledPin, HIGH);
delay(400);
digitalWrite(ledPin, LOW);
delay(400);
    }

else if (digitalRead(inputPin2) == HIGH)
   {
digitalWrite(ledPin, HIGH);
delay(100);
digitalWrite(ledPin, LOW);
delay(100);
    }

}
```

مراجعه سريعة على ما تعلمناه فى الامثله السابقه

void setup ( ) *{.....}*	الدالـه المسـؤله عـن ضـبط اعـدادات المتحكمه و تحديد المداخل و المخارج
void loop ( ) *{.....}*	يثم كتابه البرنـامج الرئيسـى داخـل هذه الداله و يقـوم البرنـامج باعـاده نفسـه مره تلـو الاخـرى كلمـا اتم تنفيـذ جميـع الخطوات
int name = value;	امر يستخدم لتعريف متغير بقيمه معينه
cons int name = value; **example:** *const int led = 13;*	امر يستخدم لتعريف ثابت غير قابل للتغير يشبه الأمر السابق مع أضافه const
pinMode (pin number, State); **example1:** pinMode(11,INPUT); **example2:** pinMode(led,OUTPUT);	الامر المسؤل عن تحديد وظيفه كل pin و ضـبطها للعمـل كمـدخل ام كمخـرج INPUT or OUTPUT
digitalWrite (pin number, State); **example:** *digitalWrite(13,HIGH);*	الامـر المسـؤل عـن اخـراج جهـد علـى احدى الـ pin بصوره رقميه HIGH (1) or LOW(0)
digitalRead (pin number); **example:** *digitalRead(4);*	الامر المسؤل عن قراءه الجهد المطبق على احدى الـpin بصوره رقميه حيث يقـوم بقيـاس الجهـد (اذا كـان 5 فولـت حتى 3.8 فولت يعتبر HIGH و اذا كان

ما بين صفر حتى 1.8 فولت يعتبر (LOW	
تأخير بزمن محدد : يستعمل هذا الامر في تحديد المده الزمنيه للامر الذى يسبقه في التنفيذ	*delay (time);* **example1:** *delay(100);* **example2:** *delay(1000);*
(اذا حدث ...او اذا حدث.. افعل .....) جمله شرطيه تستخدم لتنفيذ امر ما عندما يتحقق شرط معين و يمكن استخدم الشروط التاليه: == يساوي != لا يساوي <= أقل من أو يساوي >= أكبر من أو يساوي < أقل من > أكبر من	if (the condition) {…..what to do} else if (another condition) {…what to do} else (last condition) {….what to do}

اشهر أنواع المتغيرات التي يمكن استخدامها في اردوينو كالتالي:

النوع	مثال	القميه
Integers	int led = 13;	أي عدد صحيح من سالب 32768 حتى موجب 32768
Float	float sensor = 12.5;	يستخدم مع الأعداد التي تحتوى كسور
character	char name= hello;	حروف نصيه أو كلمات
long	long variable = 99999999;	رقم طويل جدا يتراوح بين سالب 2,147,483,648 حتى موجب نفس الرقم
byte	byte number= 44;	رقم صغير يتراوح بين (0) إلى 255

لمزيد من المعلومات عن أنواع المتغيرات التي يمكن استخدامها داخل برامج اردوينو قم بزياه الصفحة التالية

http://www.arduino.cc/en/Reference/VariableDeclaration