كلية الهندسة
قسم هندسة الميكاترونيك

المتحكمات الصغرية و النظم المضمنة

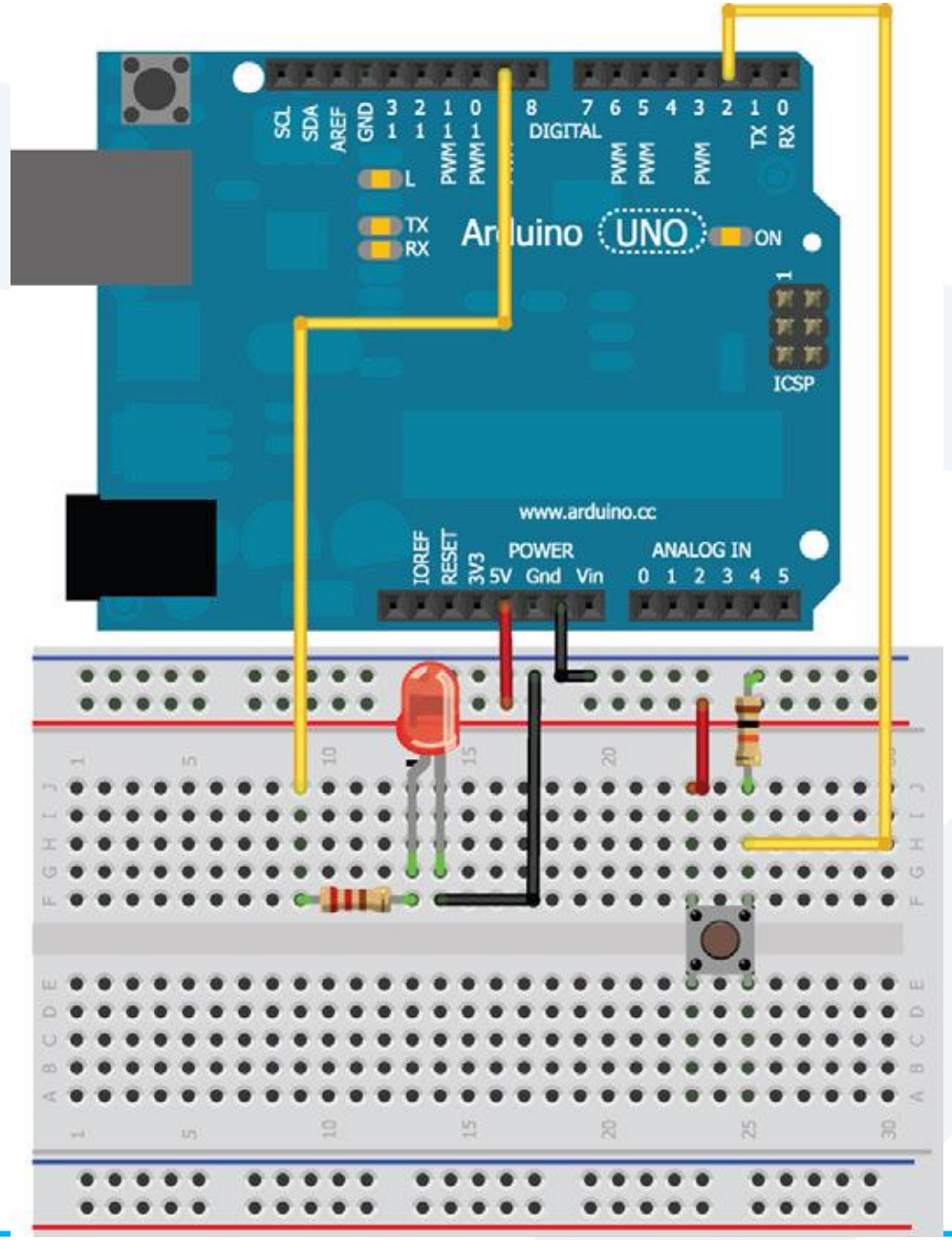
محاضرة 10

Arduino

د. فادي متوج

قراءة المدخلات الرقمية

- بعد أن درسنا كل من الخرج الرقمي والتشابهي. سنقوم بدراسة كيفية قراءة المدخلات الرقمية ، مثل المفاتيح والأزرار.



توصيل السويتش باستخدام مقاومة Pulldown

```
const int LED=9;          //The LED is connected to pin 9
const int BUTTON=2;   //The Button is connected to pin 2
void setup() {
pinMode (LED, OUTPUT);       //Set the LED pin as an output
pinMode (BUTTON, INPUT);   //Set button as input (not required
}
void loop(){
if (digitalRead(BUTTON) == LOW)
  {
    digitalWrite(LED, LOW);
   }
else
  {digitalWrite(LED, HIGH); }
}
```

قراءة قيم الحساسات التشابهية

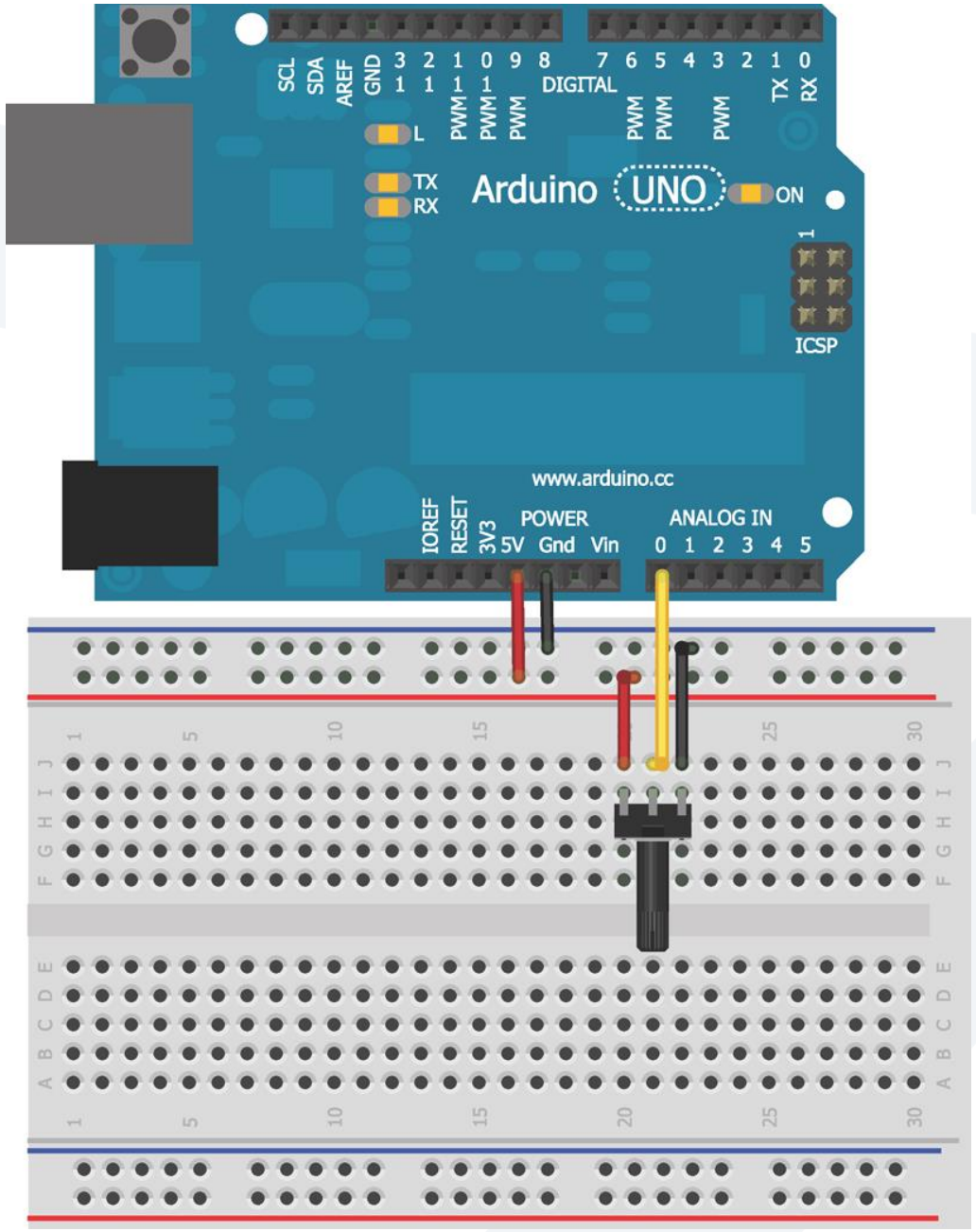
- العالم من حولنا تشابهي. على الرغم من أننا قد نسمع أن العالم "يتحول إلى عالم رقمي" ، فإن غالبية الظواهر التي يمكن ملاحظتها في عالمنا هي دائمًا تشابهية في طبيعتها.
- يمكن أن نذكر عدة أمثلة عن الإشارات التشابهية مثل شدة إشعاع الشمس ، أو درجة حرارة مياه المحيط ، أو تركيز الملوثات في الهواء.
- للتعامل مع هذه الإشارات التشابهية يجب تحويلها إلى قيم رقمية يمكن معالجتها باستخدام متحكم مثل Arduino

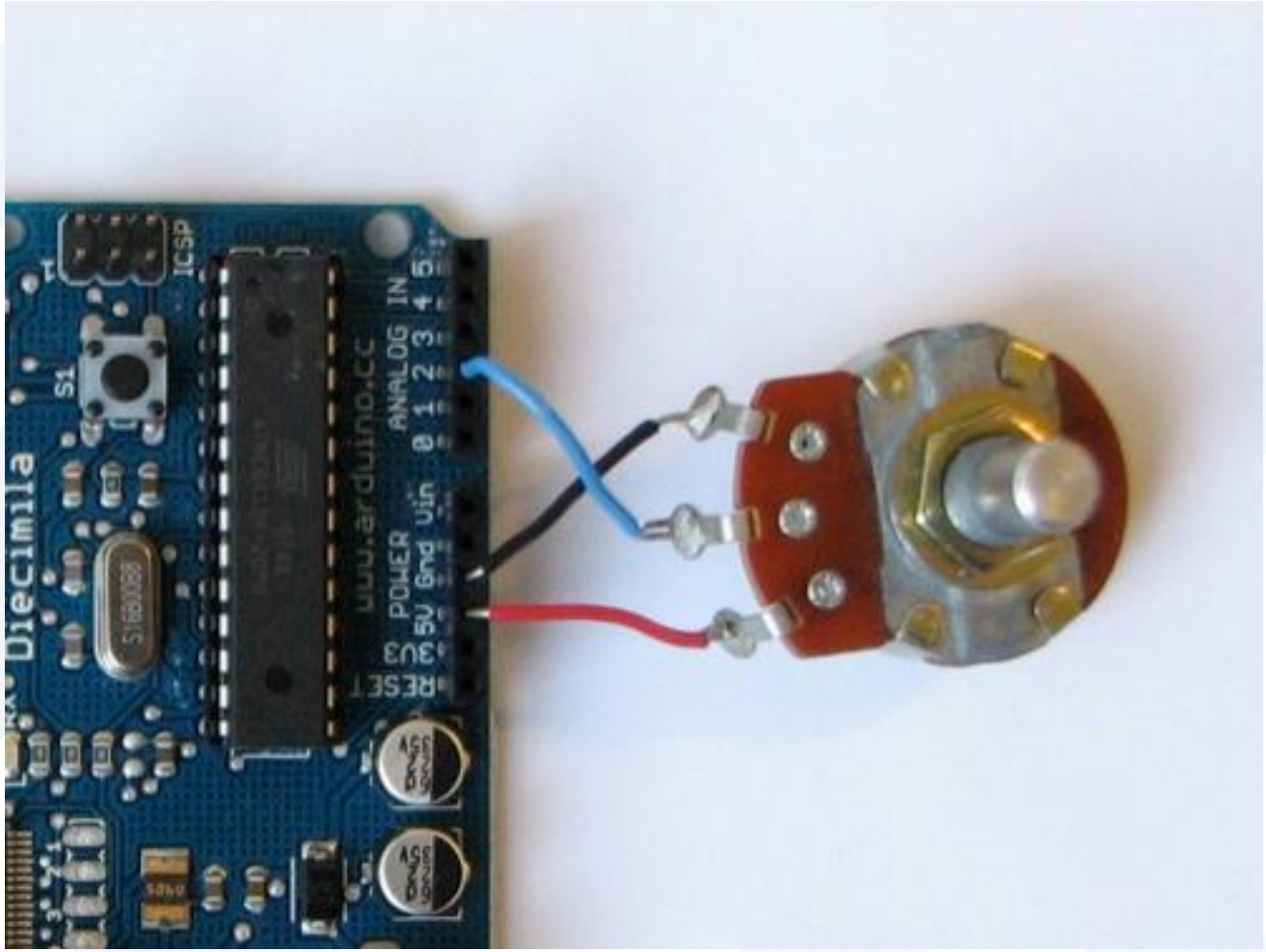
تحويل الإشارة التشابهية إلى رقمية

- يمكننا استخدام المحول التشابهي إلى رقمي (ADC) المتضمن في Arduino لتحويل قيم الجهد التشابهي إلى تمثيل رقمي يمكننا العمل معه.
- في حالة Arduino Uno، يوجد ADC بدقة 10بت لإجراء التحويل من تشابهي إلى رقمي.
- "10 بت" يعني أن ADC يمكنه تقسيم إشارة تشابهية إلى 1024 قيمة مختلفة، وبالتالي يمكن لـ Arduino تحديد قيمة رقمية من **0** إلى **1023** لأي قيمة تشابهية نعطيها له.
  - يؤدي وضع 0 فولت على دخل ADC إلى إرجاع قيمة 0 رقمي،
  - يؤدي وضع 2.5 فولت على دخل ADC إلى إرجاع قيمة 512 (نصف 1024)
  - يؤدي وضع 5 فولت على دخل ADC إلى إرجاع قيمة 1023.

مثال1 : قراءة قيمة مقاومة متغيرة (إشارة تشابهية)

- المقاومات المتغيرة هي مقسمات جهد متغيرة تأتي بأحجام وأشكال كثيرة، لكن لديها جميعًا ثلاثة أرجل.
  - نقوم بتوصيل أحد الأرجل الخارجية بالأرضي والآخر بـ 5 فولت.
  - المقاومات المتغيرة متناظرة ، لذلك لا يهم الجانب الذي توصل به 5 فولت والأرضي.
  - نقوم بتوصيل الرجل الوسطى بأحد أرجل الدخل التشابهي (A0 مثلاً) على Arduino.


SCL
SDA
AREF
GND
3 2 1 0 9 8
1 1 1 1
PWM PWM PWM PWM
DIGITAL
7 6 5 4 3 2 1 0
PWM PWM PWM
TX RX
L
TX
RX
Arduino UNO
ON
ICSP
www.arduino.cc
IOREF
RESET
3V3
POWER
5V Gnd Vin
ANALOG IN
0 1 2 3 4 5


ICSP
S1
ANALOG IN
0 1 2 3 4 5
www.arduino.cc
POWER
5V Gnd Vin
3V3
RESET

- أثناء قيامنا بتحريك مقبض المقاومة المتغيرة، نقوم بتغيير الجهد الذي نقوم بإدخاله في المدخل التشابهي A0 بين 0 فولت و 5 فولت .
- سنستخدم وظيفة **الاتصال التسلسلي** في Arduino لطباعة قيمة خرج المحول ADC للمقاومة المتغيرة على جهاز الكمبيوتر الخاص بنا أثناء تغيير قيمتها.
- سنستخدم التابع **analogRead ()** لقراءة قيمة رجل الدخل التشابهية المتصلة ببورد Arduino
- سنستخدم التابع **Serial.println ()** لطباعة القيمة المقروءة على شاشة Arduino IDE حيث نقوم بفتح نافذة الإظهار عبر النقر على زر serial monitor.
- يجب أن نقوم بتهيئة الواجهة التسلسلية للكمبيوتر وذلك باستخدام تعليمة Serial.begin () ضمن التابع setup().
- يأخذ التابع Serial.begin () معامل واحد يحدد سرعة الاتصال أو معدل نقل البيانات .
- يحدد معدل نقل البيانات عدد البتات التي يتم نقلها في الثانية.
- تمكننا معدلات النقل الأسرع من إرسال المزيد من البيانات في وقت أقل ، ولكنها قد تؤدي أيضًا إلى حدوث أخطاء في الإرسال في بعض أنظمة الاتصال.

```
//Potentiometer Reading Program
const int POT=0;   //Pot on analog pin 0
int val = 0; //variable to hold the analog reading from the POT
void setup()
{
Serial.begin(9600);
}
void loop()
{
val = analogRead(POT);
Serial.println(val);
delay(500);
}
```

pot | Arduino 1.0.3

File Edit Sketch Tools Help

pot §

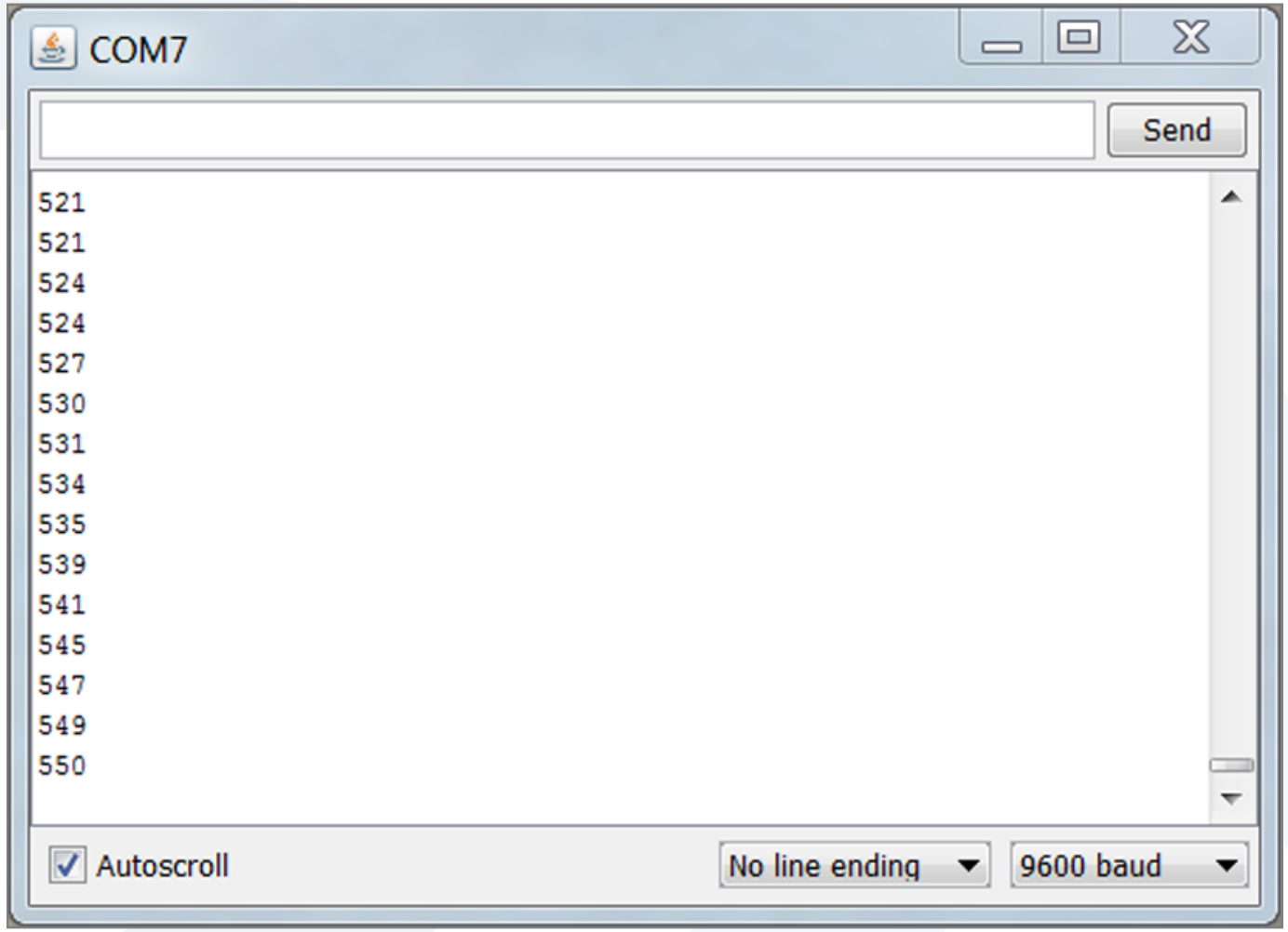
```
//Potentiometer Reading Program

const int POT=0;  //Pot on Analog Pin 0
int val = 0;      //variable to hold the analog reading from the POT

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

void loop()
{
  val = analogRead(POT);
  Serial.println(val);
  delay(500);
}
```

COM7

Send

```
521
521
524
524
527
530
531
534
535
539
541
545
547
549
550
```

Autoscroll | No line ending | 9600 baud


COM7
Send
521
521
524
524
527
530
531
534
535
539
541
545
547
549
550
Autoscroll
No line ending
9600 baud

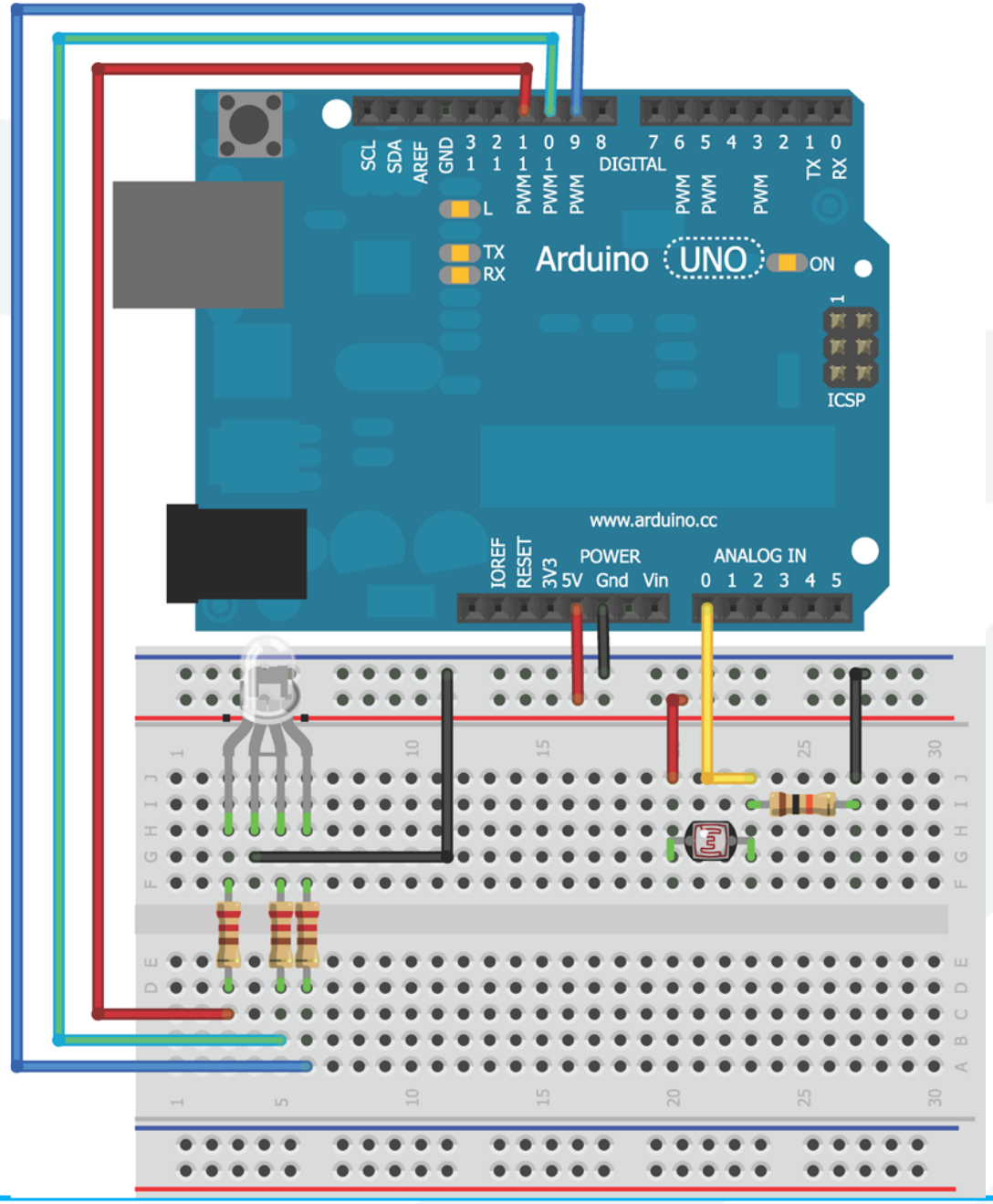
مثال2 : المقاومات الضوئية

- تغير المقاومات الضوئية مقاومتها اعتمادًا على كمية الضوء التي تسقط عليها.
- مثلاً المقاومة الضوئية 200 kΩ عندما تكون في الظلام الدامس، تكون مقاومتها حوالي 200kΩ أما عند التشبع بالضوء، تنخفض المقاومة تقريبًا إلى الصفر.
- يمكننا استخدام المقاومة الضوئية لبناء نظام إنارة ذكي (يجب أن يصبح الضوء أكثر سطوعًا حين تصبح الغرفة مظلمة والعكس صحيح).

بناء نظام إنارة ذكي باستخدام المقاومة الضوئية

- في البداية نحدد القيم المقابلة لمستوى إنارة الغرفة في السطوع الكامل و الظلام التام.
- يمكن أن نستخدم برنامج serial monitor القراءة التسلسلية الذي مر معنا سابقاً و الذي يمكننا من خلاله تحديد قيم الإنارة عندما تكون غرفتنا في السطوع الكامل أو الظلام التام.
- على سبيل المثال، وجدنا أن الغرفة المظلمة تقابل قراءة قيمة 200 رقمي وأن الغرفة المضيئة تمامًا تقابل قراءة قيمة 900 رقمي.
- ستختلف هذه القيم بناءً على ظروف الإضاءة الخاصة بكل مكان يراد تصميم نظام الإنارة له وقيمة المقاومة الضوئية التي نستخدمها.

توصيل الدارة



استخدام الدخل التشابهي للتحكم في الخرج التشابهي

- يمكننا استخدام الأمر analogWrite () للتحكم بشدة سطوع LED الذي يمثل نظام الإنارة لدينا.
- يقبل الأمر analogWrite () قيم بين 0 و 255 فقط ، بينما تقوم تعليمة القراءة التشابهية analogRead() بإرجاع قيم تصل إلى 1023 كقيمة عظمى .
- تحتوي لغة برمجة Arduino على تابعين مفيدين في الربط بين مجالين من القيم: تابع map () و تابع constrain() .
- يستخدم التابع map () على النحو التالي: output = map(value, fromLow, fromHigh, toLow, toHigh)
- value : هي المعلومات التي نبدأ بها و الواقعة في المجال الأول (بين 0 و 1023). في حالتنا، هذه هي أحدث قراءة من الإدخال التشابهي و التي تمثل مستوى إضاءة الغرفة.
- fromLow و fromHigh : هي حدود الإدخال (حدود المجال الأول). و هي القيم التي وجدنا أنها تتوافق مع الحد الأدنى والحد الأقصى للسطوع في غرفتنا. على سبيل المثال ، كانت 200 و 900 على الترتيب.
- toLow و to High : هي حدود المجال الثاني و التي تمثل القيم التي نريد ربط القيمة من المجال الأول بها و التي تمثل المستوى الذي يجب أن يضيء به نظام الإنارة. و نظرًا لأن تعليمة analogWrite () تتوقع أن تأخذ قيمة بين 0 و 255 ، فإننا نستخدمها كحدود لهذا المجال.
- في مشروع نظام الإنارة الذكي كلما ازداد ظلام الغرفة يجب أن يعمل النظام على زيادة شدة إضاءة الـ LED بحيث يصبح أكثر سطوعاً . لذلك، عندما يكون الإدخال من ADC ذو قيمة منخفضة، فإننا نريد أن يكون الخرج على الـ LED عالي القيمة، والعكس صحيح.

```
const int RLED=9;      //Red LED on pin 9 (PWM)
const int LIGHT=0;      //Light Sensor on analog pin 0
const int MIN_LIGHT=200;    //Minimum expected light value
const int MAX_LIGHT=900;   //Maximum Expected Light value
int val = 0;          //variable to hold the analog reading
void setup()
{
pinMode(RLED, OUTPUT);      //Set LED pin as output
}
void loop()
{
val = analogRead(LIGHT);              //Read the light sensor
val = map(val, MIN_LIGHT, MAX_LIGHT, 255, 0);   //Map the light reading
val = constrain(val, 0, 255);         //Constrain light value
analogWrite(RLED, val);               //Control the LED
}
```